UNIVERSIDADE FEDERAL DE SANTA CATARINA
DEPARTAMENTO DE COMPUTAÇÃO

Luan Carlos da Silva Casagrande

COMPARATIVE STUDY OF IMAGE TEXTURE ANALYSIS AND MACHINE LEARNING METHODS FOR CLASSIFICATION OF *PHRAGMITES AUSTRALIS* USING TRUE-COLOR HIGH RESOLUTION IMAGES

Araranguá

2017

Luan Carlos da Silva Casagrande

COMPARATIVE STUDY OF IMAGE TEXTURE ANALYSIS AND MACHINE LEARNING METHODS FOR CLASSIFICATION OF *PHRAGMITES AUSTRALIS* USING TRUE-COLOR HIGH RESOLUTION IMAGES

Trabalho de Conclusão de curso submetido à Universidade Federal de Santa Catarina, como parte dos requisitos necessários para a obtenção do Grau de Bacharel em Engenharia de Computação.
Orientador: Prof. Dr. Gustavo Mello Machado
Coorientador: Prof. Dr. Sathishkumar Samiappan

Araranguá

2017

Ficha de identificação da obra elaborada pelo autor,
através do Programa de Geração Automática da Biblioteca Universitária da UFSC.

da Silva Casagrande, Luan Carlos
Comparative study of image texture analysis and machine learning methods for classification of Phragmites australis using true-color high resolution images / Luan Carlos da Silva Casagrande ; orientador, Gustavo Mello Machado, coorientador, Sathishkumar Samiappan, 2017.
150 p.

Trabalho de Conclusão de Curso (graduação) - Universidade Federal de Santa Catarina, Campus Araranguá, Graduação em Engenharia de Computação, Araranguá, 2017.

Inclui referências.

1. Engenharia de Computação. 2. Image Texture Analysis. 3. Machine Learning. 4. Phragmites. 5. Low altitude remote sensing. I. Mello Machado, Gustavo. II. Samiappan, Sathishkumar. III. Universidade Federal de Santa Catarina. Graduação em Engenharia de Computação. IV. Título.

Luan Carlos da Silva Casagrande

COMPARATIVE STUDY OF IMAGE TEXTURE ANALYSIS AND MACHINE LEARNING METHODS FOR CLASSIFICATION OF PHRAGMITES AUSTRALIS USING TRUE-COLOR HIGH RESOLUTION IMAGES

Este Trabalho de Conclusão de Curso foi julgado aprovado para a obtenção do Título de "Bacharel em Engenharia de Computação" e aprovado em sua forma final pela Universidade Federal de Santa Catarina.

Araranguá, 07 de dezembro de 2017.

Prof[a]. Eliane Pozzebon, Dra.
Coordenadora do Curso

Banca Examinadora:

Prof. Gustavo Mello Machado, Dr.
Universidade Federal de Santa Catarina

Prof. Alexandre Leopoldo Gonçalves, Dr.
Universidade Federal de Santa Catarina

Prof. Anderson Luiz Fernandes Perez, Dr.
Universidade Federal de Santa Catarina

Prof. Sathishkumar Samiappan, Dr.
Mississippi State University

ACKNOWLEDGMENT

We do not accomplish our achievements without the support of people, so I would like to thank some of them. Firstly, I would like to thank my advisor Dr. Gustavo Mello Machado and my co-advisor Dr. Sathishkumar Samiappan for their advice and assistance in completing this work. I am very grateful for everything that they have been doing for me. This work would not be possible without them.

I would also like to thank Dr. Robert Moorhead for giving me the summer internship opportunity at Geosystem Research Institute in Mississippi. During this internship I had my first official experience with remote sensing and image processing, so I don't have words to express how thankful I am this amazing experience.

Additionally, I would like to thank Gray Turnage, Lee Hathcock, and Sean Meacham from Geosystem Research Institute for assisting me in getting the data, and Dr. Roderval Marcelino and Dr. Vilson Gruber from Laboratório de Pesquisa Aplicada e Laboratório de Telecomunicações for assisting me with the necessary structure to develop this work.

Finally I thank my friends Giann Spilere Nandi and Daniel Bitencourt for their help in the revision process and last, but not least, I would like to thank my family and Bruna Lima for their continued support through everything.

RESUMO

Phragmites australis (common reed) comumente encontrada em zonas úmidas costeiras pode alterar rapidamente a ecologia por competir e superar as plantas nativas por espaço e pelos recursos. Além disso, este tipo de vegetação representa um perigo de navegação para embarcações menores, prejudicando a visibilidade ao longo do litoral e em torno de curvas e canais de rios. Os esforços de gerencialmento direcionados a plantas não nativas de *Phragmites* dependem fortemente de um mapeamento preciso das áreas invadidads. No entanto, o mapeamento de *Phragmites* representa um desafio único por diferentes razões. Identificar e mapear *Phragmites* pode ajudar os gerentes de recurso a restaurar zonas húmidas afetadas. Neste trabalho, quatro técnicas de extração de características foram testadas: gabor filters, grey level co-occurrence matrix, segmentation-based fractal texture analysis e wavelet texture analysis. Estes algoritmos foram combinados com três estruturas de rede neural artificial: multilayer perceptron, probabilistic neural network e radial basis function network. Além disso, objetivando reduzir o tempo computacional, uma implementação na *Graphics Processing Unit* do melhor método identificado foi realizada. O estudo de avaliação foi realizado com imagens adquiridas no delta de Pearl River localizado no sudeste da Louisiana e no sudoeste do Mississippi, Estados Unidos da América. Em comparação com os resultados apresentados no estado da arte, wavelet texture analysis com probabilistic neural network e segmentation-based fractal texture analysis com probabilistic neural network apresentaram melhorias em várias variáveis estatísticas como acurácia geral e o kappa. Além disso, o nível de *Phragmites agreement* aumentou considerávelmente. Nos mostramos que os erros de omissão e comissão restantes geralmente estão localizados ao longo dos limites das áreas identificadas como *Phragmites*, o que reduz os esforços desnecessários para os gerentes de recursos na busca de áreas inexistentes.

Palavras-chave: Análise de textura em imagens, aprendizado de máquina, *Phragmites*, avião não tripulado, Imagens de espectro visível.

RESUMO EXPANDIDO

INTRODUÇÃO

Phragmites australis (common reed) comumente encontrada em zonas húmidas costeiras pode alterar rapidamente a ecologia por competir e superar as plantas nativas por espaço e pelos recursos. Além disso, este tipo de vegetação representa um perigo de navegação para embarcações menores, prejudicando a visibilidade ao longo do litoral e em torno de curvas e canais de rios. Como consequência, há um grande interesse no controle desta vegetação. O principal método de controle é a aplicação de herbicidas (MARTIN; BLOSSEY, 2013; HAZELTON et al., 2014). Porém, para se aplicar qualquer método de controle é primeiro necessário identificar o local destas vegetações. Este processo representa um desafio único, pois não há padrões para novos estandes e a disseminação generalizada da *common reed* nas zonas húmidas. Diferentes métodos podem ser usados para localizar o tipo invasivo de *Phragmites*. Caminhar ao redor ou através de *Phragmites* com uma unidade de GPS para mapear manualmente esta vegetação é um dos métodos mais simples. No entanto, os ambientes das zonas húmidas têm pouca acessibilidade devido à vegetação densa, alta, emergente e profundidades variadas de água, impossibilitando o trabalho de campo (LANTZ; WANG, 2013). Sensoriamento Remoto fornece um método alternativo para a aquisição de dados que podem ajudar a reduzir custos, mão-de-obra e economizar tempo em relação ao trabalho de campo. Diferentes combinações de plataformas com sensores podem possibilitar a aquisição de dados de diferentes resoluções espaciais, espectrais e temporais. Dentre as plataformas possíveis, as mais comumente utilizadas para mapeamento de *Phragmites* são: Satélite, aeronave tripulada e aeronave não tripulada. A primeira destaca-se por sua maior capacidade de cobertura, menor dependência pós-processamento e velocidade para adquirir os dados. Vários trabalhos foram propostos com dados coletados através de satélite, como demonstrado em (GILMORE et al., 2008; LANTZ; WANG, 2013; PENGRA; JOHNSTON; LOVELAND, 2007; BOURGEAU-CHAVEZ et al., 2013; VILLA et al., 2013). A segunda plataforma se diferencia por proporcionar um melhor aproveitamento de sensores hiperespectrais, criando a possibilidade de soluções com maior resolução espectral. Becker (2009), Artigas and Pechmann (2010) e Villa et al. (2013) apresentaram trabalhos com dados coletados através desta plataforma. Já as aeronaves não tripuladas permitiram uma nova metodologia de sensoriamento remoto chamada de *low altitude remote sensing*. O interesse nessa abor-

dagem está crescendo principalmente devido ao seu custo, segurança e possibilidade de coleta de dados com alta resolução espacial. Consequentemente, recentemente, vários trabalhos propuseram novas técnicas de mapeamento de *Phragmites* com base nessa metodologia. Dentre eles, Samiappan et al. (2016b) e Casagrande et al. (2017) apresentaram soluções que podem ser considerados o estado da arte no escopo deste trabalho. Neste trabalho tentou-se responder a seguinte pergunta de pesquisa: Como desenvolver uma nova solução para mapear *Phragmites australis* usando imagens de alta resolução no espectro do visível, coletadas através de uma aeronave não tripulada, através do uso de técnicas de análise de textura e aprendizagem de máquina que consiga reduzir o erro espacial e o custo computacional?

OBJETIVOS

O objetivo geral deste trabalho é desenvolver um framework eficiente para mapeamento de *Phragmites australis* usando dados coletados por sistemas aéreos não tripulados através de um sistema de classificação fundamentando em análise de textura e aprendizagem de máquina. Para atingir tal objetivo, visa-se melhorar a precisão geral do mapeamento, reduzir o custo computacional através de uma implementação fundamentada na *General-purpose Graphics Processing Unit* e comparar as precisões de mapeamento e a eficiência computacional dos métodos propostos com o estado da arte.

METODOLOGIA

O framework proposto neste trabalho pode ser resumido em seis etapas: Antes da extração de características, extração de características fundamentadas na textura, seleção dos conjuntos, ajuste de parâmetros, classificação através de aprendizagem de máquina e a pós-classificação. Neste caso, considerando-se que as combinações precisavam ser avaliadas, uma etapa adicional foi posicionada após a pós-classificação. A etapa de ajuste dos parâmetros, a implementação de parte das soluções utilizando os recursos da *Graphics Processing Unit* e o estudo de diferentes arquiteturas de redes neurais artificiais são diferenciais deste trabalho. Na primeira etapa da solução, a informação geográfica do mosaico gerado é armazenada, a imagem é dividida em blocos de 100x100 pixels e estes blocos são convertidos de *red-green-blue* para escala de

cinza. Tal divisão de blocos resulta em aspectos positivos e negativos para o sistema. Além de reduzir o tempo computacional por não realizar uma classificação por pixel, a informação espacial dos blocos é essencial para a etapa de classificação. Porém, o uso de blocos prejudica a acurácia geral do sistema em situações específicas. Após, quatro técnicas de análise de textura foram testadas na etapa de extração de características, sendo: *Grey Level Co-occurrence Matrix*, *Gabor Filters*, *Segmentation-based Fractal Texture Analysis* e *Wavelet Texture Analysis*. Estas técnicas também foram utilizadas no trabalho de Samiappan et al. (2016a). Cada uma destas técnicas tem suas especificidades que podem contribuir ou não para o resultado final do sistema. Após, neste caso específico, os dados extraídos são divididos entre treinamento, teste e validação. Primeiro, 10% de todos os blocos são selecionados para a etapa de ajuste dos parâmetros. Os outros 90% são utilizados para validação. Destes 10%, 80% dos blocos são então utilizados para treinamento e os outros 20% para teste. Neste trabalho, três arquiteturas de rede neural artificial foram analisadas: *multilayer perceptron*, *probabilistic neural network* e *radial basis function network*. Na primeira arquitetura, utilizou-se o algoritmo de treinamento Levenberg-Marquard. Já na *radial basis function network*, utilizou-se o treinamento fundamentado em dois estágios: *k-means* para definição dos centros e o método de equações normais para otimização dos pesos. Após a etapa de definição dos conjuntos, ocorre a etapa de ajuste dos parâmetros. Neste caso, dois algoritmos foram utilizados: Algoritmo busca em grid e algoritmo genético. O primeiro foi utilizado para identificar o sigma do algoritmo *probabilistic neural network*. Já o segundo foi utilizado com as outras duas arquiteturas. Em ambos os casos, o espaço de busca foi o fator decisivo. Após este processo, ocorre então a predição de classe para os blocos não identificados e os dados salvos na etapa de pré-extração são utilizados para geração de um arquivo *shape*. Seis parâmetros foram computados para avaliar o sistema: kappa, *overall accuracy*, *agreement*, *commission error* e *omission error*. Algumas das combinações dos algoritmos cidatos foram também implementados visando a *Graphics Processing Unit*.

RESULTADOS E DISCUSSÃO

O estudo de avaliação foi realizado com imagens adquiridas no delta de Pearl River localizado no sudeste da Louisiana e no sudoeste do Mississippi, Estados Unidos da América. A área total foi dividida em

duas imagens. Os dados foram coletados por meio de uma aeronave não tripulada a prova d'água com uma Canon Rebel EOS SL1. As missões de coleta de dados foram conduzidas a 300 metros de altura, com 50% de sobreposição lateral e 70% de frente. A resolução espacial final ficou em 5cm/pixel. Para o processo de validação dos resultados, um mapa de referência foi gerado por especialistas na área através de dados coletados em solo e inspeção visual. Os resultados visuais das soluções aqui propostas foram gerados através de implementação no MATLAB. Já para a comparação entre as implementações na *Central Process Unit* e na *Graphics Processing Unit*, os algoritmos foram implementados em C. Com os valores estatísticos e os mapas gerados, pode-se observar que existe um *trade-off* entre *Phragmites agreement* e não *Phragmites agreement*. Tal fato é principalmente causado pela proporção e as instâncias utilizadas no conjunto de treinamento de cada classificador. Além disso, observou-se que as principais fontes de erro são: Transição entre *Phragmites* e não *Phragmites*, mistura de vegetações e reflexão da lux no rio. A primeira fonte de erro é causada pela abordagem de blocos utilizada. Já a segunda fonte de erro, apesar de não tão comum, já era esperada neste trabalho. O mesmo problema foi relatado em vários trabalhos relacionados. O terceiro problema aconteceu apenas com as soluções baseadas em características extraídas através das técnicas *Grey Level Co-occurrence Matrix*, *Gabor Filters* e *Segmentation-based Fractal Texture Analysis*. A granularidade apresentada por *Phragmites* foi o principal fator responsável por tal problema. Para a imagem I, os melhores resultados foram obtidos com a combinação *wavelet transform analysis* com *probabilistic neural network*. Neste caso o valor final para o kappa ficou em 0.9113 e para a *overall accuracy* ficou em 97.8%. Já para a imagem II, os melhores resultados foram obtidos com a combinação *Segmentation-based Fractal Texture Analysis* com *probabilistic neural network*. O valor final de kappa ficou em 0.9362 e para a *overall accuracy* ficou em 99.2%. Apesar destes problemas descritos, as melhores combinações apresentaram mapas com alto nível de confiabilidade. Identificou-se também que *probabilistic neural network* apresentou dominância sobre as outras estruturas em praticamente todas as combinações e que as técnicas *Segmentation-based Fractal Texture* e *wavelet transform analysis* demonstraram um maior potencial quando comparadas as outras técnicas. Dois fatos podem ser concluídos da comparação entre as implementações na *Central Process Unit* e na *Graphics Processing Unit*: Primeiro, a implementação na *Central Process Unit* conseguiu reduzir drasticamente o tempo médio por quilometro quadrado quando comparado com as soluções propostas no

MATLAB. Neste caso, o tempo final para a imagem I ficou em 10 minutos. Em segundo lugar, a implementação da *Graphics Processing Unit* para este problema é viável e reduziu consideravelmente o tempo de processamento necessário para gerar um mapa.

CONCLUSÃO

Neste trabalho se apresentou um estudo comparativo de técnicas de análise de textura e de aprendizagem de máquina para classificação de *Phragmites australis* em imagens de alta resolução no espectro do visível. Além disso, implementações na *Graphics Processing Unit* foram propostas visando reduzir o tempo computacional. Para maior consistência na comparação, os mesmos dados apresentados por Samiappan et al. (2016a) foram utilizados neste trabalho. O estudo experimental demonstrou que as combinações de *Segmentation-based Fractal Texture* ou *wavelet transform analysis* com *probabilistic neural network* resultaram em mapas com maiores níveis de confiabilidade. Além disso, o tempo computacional necessário para gerar o mapa foi reduzido drasticamente com a implementação na *Graphics Processing Unit*. Este resultado já era esperado, pois a solução proposta pode ser facilmente paralelizada.

Palavras-chave: Análise de textura em imagens, aprendizado de máquina, *Phragmites*, avião não tripulado, Imagens de espectro visível.

ABSTRACT

Phragmites australis (common reed) commonly found in the coastal wetlands can rapidly alter the ecology by outcompeting with natives for space and resources. In addition, this type of vegetation presents a navigation hazard to smaller boats by impairing visibility along shorelines and around bends of canals and rivers. Management efforts targeting non-native *Phragmites* rely heavily on accurately mapping invaded areas. However, mapping *Phragmites* represents a unique challenge for different reasons. Identifying and mapping *Phragmites* can help resource managers to restore affected wetlands. In this work, four feature extraction methods were tested: gabor filters, grey level co-occurrence matrix, segmentation-based fractal texture analysis, and wavelet texture analysis. These algorithms were combined with three artificial neural network architectures: multilayer perceptron, probabilistic neural network, and radial basis function network. In addition, aiming to reduce the computational cost, a graphics processing unit implementation of the best result was performed. Evaluation study was conducted with imagery acquired in the delta of Pearl River located in southeastern Louisiana and southwestern Mississippi, United States of America. In comparison to state-of-art results, wavelet texture analysis with probabilistic neural network and segmentation-based fractal texture analysis with probabilistic neural network presented presented improvements in several statistical variables such as overall accuracy and kappa value. Furthermore, the *Phragmites* agreement increased considerably. We show that the remaining omission and commission errors are generally located along boundaries of patches with *Phragmites*, which reduces unnecessary efforts for resource managers while searching for nonexistent patches.

Keywords: Image Texture Analysis, Machine Learning, Phragmites, low altitude remote sensing, visible spectrum imagery.

LIST OF FIGURES

Figure 1 Active and Passive techniques. The first has its own source of energy and can be used at any time. The second depends on external sources of energy, and in some cases, can be used just during daylight. 37
Figure 2 Response of 14 different wetland plant species and bandwidth of each spectral band of sensor. As related by Samiappan et al. (2016b), the imagery has three visible bands blue (480 nm), green (560 nm), and red (670 nm), as well as red-edge (720 nm) and near-infrared (840 nm) bands. Despite the near spectral signatures, the difference between plants is perceptible. 38
Figure 3 Comparison between images with different Ground Sample Distance (GSD). (a) Unmanned Aerial Vehicle (UAV) imagery with GSD of 4cm, (b) Manned aircraft with GSD of 50cm. 40
Figure 4 Representation of a possible application with 2 kernels where the first was divided into 2x3 blocks with an internal size of 3x5 threads. That is, there are 6 blocks where each block has 15 threads. 47
Figure 5 Compute Unified Device Architecture (CUDA) device memory hierarchy representation. 48
Figure 6 The study site near Pearlington, Mississippi, USA (about 2600 acres) selected for developing and evaluating techniques to map invasive *Phragmites*. 51
Figure 7 (I) Desert Island - total area of approximately 574ha ... 52
Figure 8 (II) Browns Island - total area of approximately 489.35ha 52
Figure 9 Waterproof Altavian Nova Unmanned Aerial System (UAS). 53
Figure 10 Ground reference for study area I (*Phragmites* locations are outlined in red) (top). Ground reference patches of *Phragmites* determined by walking along the patch boundary and determining vertices using a handheld Global Positioning System (GPS) unit (Bottom-left). Digitized boundaries – DIG from visual inspection (Bottom-right). 55
Figure 11 Segment from original set I represented through blocks. 57
Figure 12 Segment from Site I representing *Phragmites* object. ... 58
Figure 13 Image blocks examples (top), representing the five most common object categories in the study area, and their respective

GLCM signatures (bottom) concerning 22 statistical properties... 61
Figure 14 Features from image blocks representing the five most common objects. From top to down: *Phragmites*, trees, soil, river, and highway..... 63
Figure 15 Image blocks examples (top), representing the five most common object categories in the study area, and their respective Segmentation-based Fractal Texture Analysis (SFTA) features (bottom) calculated using $n_t = 8$..... 65
Figure 16 One level of Haar wavelet transform applied to (a) example block with 100×100 pixels. The output (b) is divided into four sub-regions with 50×50 pixels each. Top-left (LL) shows the approximation values, top-right (LH) shows details in horizontal, bottom-left (HL) shows details in vertical, and bottom-right (HH) shows details in diagonal..... 66
Figure 17 Image blocks examples (top), representing the five most common object categories in the study area, and their respective wavelet texture signatures (bottom) concerning four statistical properties (mean, entropy, standard deviation, and energy value) for each sub-band (LL, LH, HL, and HH)..... 68
Figure 18 Biological Neuron..... 74
Figure 19 Perceptron representation. In this image, y is the output signal, x_i is the value of the ith connection, w_i is the weight associated to the ith connection, and b is the bias associated..... 75
Figure 20 Multilayer Perceptron (MLP) fully connected architecture example. In this case, the architecture is composed of the input, one hidden, and output layer..... 76
Figure 21 Illustration of our Probabilistic Neural Network (PNN)'s architecture for classification. The input layer has n nodes, one for each input parameter. The pattern layer is fully connected to the input layer, and its nodes represent the training cases. The summation layer averages the results from pattern layer for each of the two categories (positive P or negative NP). And the output layer decides to which category (i.e., with or without *Phragmites*) the input corresponds..... 79
Figure 22 Output map from PNN using data output from Wavelet Texture Analysis (WTA) with four different σ for the set II..... 80
Figure 23 Illustration of our Radial Basis Function Network (RBFN)'s overall architecture. The input layer has n nodes, one for each input parameter. The hidden layer, here identified as radial basis

function layer, is fully connected to the input layer, and its nodes represent the identified centroids. This layer is composed of radial basis function neurons that is a specific type of artificial neuron. The output sums the weighted outputs from the hidden layer for each category (in this case, just one category with possible classes: *Phragmites* or not *Phragmites*). 82
Figure 24 Output map from RBFN using data output from WTA with four different k for the set I. 85
Figure 25 Genetic Algorithm-Artificial Neural Network algorithm flowchart composed by eight process and two decision. Three from these eight process are specific from Genetic Algorithm (GA) algorithm: selection, crossover, and mutation. 89
Figure 26 Individual general structure. The input layer is represented by the chromosome I. The hidden layers are represented by the chromosome H and can range from 1 to n where n represents the number of hidden layers. The chromosome H represents the output layer. Each chromosome type represents different features from the architecture. 90
Figure 27 Comparison between a river sunlight reflection (a) and *Phragmites* (b) blocks. 102
Figure 28 Site I assessment for PNN classification using Grey Level Co-occurrence Matrix (GLCM) texture features. 103
Figure 29 Site I assessment for MLP classification using GLCM texture features. 104
Figure 30 Site I assessment for RBFN classification using GLCM texture feature. 104
Figure 31 Site II assessment for RBFN classification using GLCM texture features. 105
Figure 32 Site I assessment for PNN classification using texture features based on Gabor Filter (GF). 107
Figure 33 Site I assessment for PNN classification using texture features based on GF. 108
Figure 34 Site I assessment for RBFN classification using SFTA texture features. 110
Figure 35 Site II assessment for RBFN classification using SFTA. 111
Figure 36 Site II assessment for PNN classification using SFTA texture features. 112
Figure 37 Site I assessment for PNN classification using WTA. 114
Figure 38 Site I assessment for MLP classification using WTA. 114

Figure 39 Site II assessment for PNN classification using WTA... 115
Figure 40 Site I assessment for k-nearest neighbors (k-NN) classification, where *Phragmites* agreement is outlined in red, omission cases outlined in green, and commission cases in blue..... 119
Figure 41 Site II assessment for k-NN classification, where *Phragmites* agreement is outlined in red, omission cases outlined in green, and commission cases in blue..... 121
Figure 42 Site I assessment for MLP classification using texture features based on GF..... 141
Figure 43 Site II assessment for PNN classification using SFTA texture features..... 141
Figure 44 Site I assessment for MLP classification using SFTA texture features..... 142
Figure 45 Site I assessment for RBFN classification using WTA... 142
Figure 46 Site II assessment for MLP classification using GLCM.. 143
Figure 47 Site II assessment for MLP classification using GF..... 144
Figure 48 Site II assessment for MLP classification using SFTA... 145
Figure 49 Site II assessment for MLP classification using WTA... 146
Figure 50 Site II assessment for RBFN classification using GLCM. 147
Figure 51 Site II assessment for RBFN classification using WTA.. 148

LIST OF TABLES

Table 1 GLCM Features 60
Table 2 GLCM Variables 61
Table 3 Wavelet Features 67
Table 4 PNN's search space totaling 90 combinations. 95
Table 5 RBFN's search space. 95
Table 6 Search Space for MLP experiment totalling 100 combinations. 96
Table 7 Results from MLP experiment. 96
Table 8 Search space for Artificial Neural Network (ANN)-MLP. 97
Table 9 Confusion Matrix for Binary Classifiers, where P represents *Phragmites*, NP not *Phragmites*, TP is the number of true positive cases, FP is the number of false positive, FN is the number of false negative, and TN is the number of true negative. 98
Table 10 CPU Experimental Setup. 100
Table 11 GPU Experimental Setup. 100
Table 12 Site I - GLCM Texture Features. 101
Table 13 Site II - GLCM Texture Features. 101
Table 14 Site I - Texture Features Based on GF. 106
Table 15 Site II - Texture Features Based on GF. 106
Table 16 Site I - SFTA Texture Features. 109
Table 17 Site I - SFTA Texture Features. 109
Table 18 Site I - WTA. PNN-2 column represents Casagrande et al. (2017) results. 113
Table 19 Site II - WTA. PNN-2 column represents Casagrande et al. (2017) results. 113
Table 20 Learning algorithms assessments for Site I 116
Table 21 Learning algorithms assessments for Site II 116
Table 22 Central Process Unit (CPU) Time 117
Table 23 Graphics Processing Unit (GPU) Time 117
Table 24 CPU Hyperparameter Tuning and Map Generation steps 118
Table 25 GPU Hyperparameter Tuning and Map Generation steps for PNN solutions 118

LIST OF ABBREVIATIONS AND ACRONYMS

GPS	Global Positioning System	28
RS	Remote Sensing	29
LARS	Low Altitude Remote Sensing	29
UAS	Unmanned Aerial System	29
GF	Gabor Filters	29
GLCM	Grey Level Co-occurrence Matrix	29
SFTA	Segmentation-based Fractal Texture Analysis	29
WTA	Wavelet Texture Analysis	29
SVM	Support Vector Machine	29
PNN	Probabilistic Neural Network	29
GPGPU	General-purpose Graphics Processing Unit	30
ANN	Artificial Neural Network	30
GPU	Graphics Processing Unit	30
CPU	Central Process Unit	32
CUDA	Compute Unified Device Architecture	32
GA	Genetic Algorithm	32
GSD	Ground sample distance	40
UAV	Unmanned Aerial Vehicle	40
TTBD	Two-threshold Binary Decomposition	45
AMD	Advanced Micro Devices	46
OpenCL	Open Computing Language	46
DRAM	Dynamic random access memory	47
USA	United States of America	51
ppt	Practical Salinity Scale	51
GRM	Ground Reference Map	54
RBFN	Radial Basis Function Network	71
MLP	Multilayer Perceptron	71
MSE	Mean Square Error	72
LM	Levenberg-Marquard	77

CONTENTS

1 INTRODUCTION 27
1.1 OBJECTIVES 30
1.1.1 General 30
1.1.2 Specifics 30
1.2 HYPHOTESIS 30
1.3 JUSTIFICATION 30
1.4 METHODOLOGY 32
1.5 WORK STRUCTURE 33
2 BACKGROUND 35
2.1 REMOTE SENSING 35
2.1.1 Historic Context 36
2.1.2 Electromagnetic Energy 36
2.1.3 Energy Interaction with the Earth's Surface 37
2.1.4 Digital Images 39
2.1.5 Sensors and Platforms 40
2.2 PATTERN RECOGNITION 42
2.3 GENERAL-PURPOSE GRAPHICS PROCESSING UNIT 46
3 STUDY AREA 51
3.1 IMAGE ACQUISITION 53
3.1.1 Ground Reference Map 54
4 TEXTURE FEATURE ANALYSIS 57
4.1 FEATURE EXTRACTION 58
4.1.1 Grey Level Co-occurrence Matrix 59
4.1.2 Gabor Filters 62
4.1.3 Segmentation-based Fractal Texture Analysis 64
4.1.4 Wavelet Texture Analysis 65
4.1.5 GPU Implementation 67
5 CLASSIFICATION SYSTEM 71
5.1 PRE-CLASSIFICATION 71
5.2 SUPORT VECTOR MACHINE 72
5.3 CLASSIFICATION BASED ON ANN 73
5.3.1 Multilayer Perceptron 75
5.3.1.1 Levenberg-Marquard Learning Algorithm 77
5.3.2 Probabilistic Neural Network 78
5.3.3 Radial Basis Function Network 81
5.3.3.1 Radial Basis Function Network architecture 82
5.3.3.2 RBFN Learning 83

5.4 HYPERPARAMETER TUNING 85
5.4.1 Grid-Search Algorithm 86
5.4.2 Genetic Algorithm 87
5.4.2.1 Genetic Algorithm and ANN-MLP 87
5.4.2.2 Optimization Algorithm 88
5.4.3 GPU Implementation 91
6 RESULTS AND DISCUSSIONS 95
6.1 IMPLEMENTATION SPECIFICATIONS 95
6.1.1 Hyperparameter Tuning Search Space 95
6.1.2 System Assessment 97
6.1.2.1 Accuracy Assessment Methodology 97
6.1.3 Computational Time Assessment Methodology 99
6.1.4 Experimental Setup 99
6.2 ACCURACY ASSESSMENT 100
6.2.1 Grey Level Co-Occurrence Matrix 101
6.2.2 Gabor Filters 106
6.2.3 Segmentation-based Fractal Texture Analysis 109
6.2.4 Wavelet Texture Analysis 113
6.3 COMPARATIVE STUDY WITH OTHER TECHNIQUES 115
6.4 CPU X GPU ASSESSMENT 117
6.5 DISCUSSIONS 118
7 CONCLUSION 123
References 125
ANNEX A – Gabor Extracted Features Examles 141

1 INTRODUCTION

Phragmites australis (Cav.) Trin. ex Steud. (common reed) is a perennial grass species that is found on every continent except Antarctica. In the United States, it is found in the 48 contiguous states (M LAPIN B, 1994). This vegetation type is hollow, rigid and it has woody stems. It has a vigorous root system called rhizomes and can grow up to 4.8m with an average height of 3.6m in brackish and freshwater wetlands. It has the potential for propagation through sexual reproduction, seed germination, or vegetative reproduction through the rhizomes (KETTENRING et al., 2009).

According to historical records (NIERING; WARREN; WEYMOUTH, 1977; GOMAN; WELLS, 2000), *Phragmites* is native to the North American continent. However, its distribution and abundance has increased dramatically over the past 150 years, mainly because of human disturbance of the landscape and introduction of novel genetic lineages (SALTONSTALL, 2002, 2003). As reported by (SALTONSTALL, 2002), there are 14 *Phragmites* haplotypes, where 11 are native and 3 non-natives (Haplotypes M, L and I). These non-natives were most likely introduced from populations originating in Europe or Asia.

Native *Phragmites* were not historically found on the Gulf Coast of the US (SALTONSTALL et al., 2005). In this region, the non-native haplotype I is predominant (SALTONSTALL, 2002, 2003). Unlike its native counterpart, invasive *Phragmites* creates dense, near mono-specific stands that outcompete native vegetation (KETTENRING et al., 2009). It alters considerably the invaded environments, posing a threat to biodiversity, water resources, and human and animal well-being. *Phragmites* invasions are frequently associated with decreases in plant biodiversity (BERTNESS; EWANCHUK; SILLIMAN, 2002), declines in habitat quality for fish and wildlife (CHAMBERS; MEYERSON; DIBBLE, 2012), disruptions to biogeochemical cycles (MEYERSON; CHAMBERS; VOGT, 1999) and other ecosystem services.

In addition to these biological impacts, this invasive plant is responsible for social impacts, such as: it is a navigation hazard by reducing visibility; it is a fire danger for nearby residents mainly by the plant material that dies during each fall creating concentrations of tinder-dry vegetation; it reduces the value of the property by infesting the landowners area; it blocks access to the water for swimming, fishing, and other recreation activities; and it obscures views of landowners, nearby residents, and visitors, among others. As a consequence, there

is a growing interest in controlling this type of invasive vegetation. Between 2005 and 2009, private and public conservation organizations in the United States spent more than 4.6 million per year on *Phragmites* removal (MARTIN; BLOSSEY, 2013).

The main tools used by the resource managers to reduce or mitigate the impact of invasive *Phragmites* are mowing, cutting, burning, grazing, and mainly herbicides (MARTIN; BLOSSEY, 2013; HAZELTON et al., 2014). However, in order to apply any control methods, it is first necessary to map the locations of this vegetation quickly and accurately. This process of precise mapping presents a unique challenge, mainly because there is no pattern to spawn new stands and the pervasive spread of common reed in wetlands.

Multiple methods can be used to map the vegetation. Walking around or through *Phragmites* stand with a GPS unit to manually map this vegetation is one of the simplest methods. However, wetland environments have poor accessibility due to dense, tall, emergent vegetation and varying depths of water, making fieldwork impossible or impractical for large areas (LANTZ; WANG, 2013). In addition, it can be dangerous to field crew members due to the cited aspects, dangerous wildlife, or insect-borne diseases. Considering the fast propagation and the actual spreading stage of *Phragmites*, this process of manually mapping is impractical. As a consequence of all challenges involving this mapping method, several authors proposed the use of remote sensing platform to solve this problem.

Remote Sensing (RS) can be defined as the practice of deriving information about the Earth's land and water surfaces using images acquired from an overhead perspective, using electromagnetic radiation in more than one spectrum region, reflected or emitted from the Earth's surface (CAMPBELL, 1996). It provides an alternative method for acquisition of data which can help to reduce costs, labor, and save time relative to field work. Different remote sensing platforms can provide the possibility to acquire data from different spatial and temporal resolutions. The spatial resolution, that is the on the ground size of a single pixel, can range from sub-centimeters to several kilometers. High-resolution aerial images may be more feasible to discern individual landscape features, such as scattered weeds (LAMB; BROWN, 2001). The temporal resolution that is the time between image acquisition events, can vary between years, weeks, days, or less. Each remote sensing platform has its own limitations in relation to the time resolution, and, considering that this plant can colonize even small patches of disturbed soils very quickly (KETTENRING et al., 2009), a small temporal

resolution is important for the mapping process as well. All these remote sensing characteristics create the possibility of a wide range of applications for mapping this specific type of vegetation.

The most common remote sensing platforms are the satellite, manned aircraft, and unmanned aircraft. The analysis based on imagery acquired by high-resolution sensors mounted on planes or satellites can be relatively expensive (ANDERSON; GASTON, 2013). Poor spatial resolution of satellite or aircraft captured imagery restricts the ability to delineate and map small *Phragmites* patches. Additionally, for free satellite imagery, the rate of orbit and time to market for imagery may not be feasible for resource manager goals. Consequently, reestablishment of a *Phragmites* stands after management efforts may go unnoticed until satellite imagery is updated. Manned aircraft is an alternative to satellites, but can be prone to pilot error during image collection. Moreover, it can be costly and may have a lower spatial resolution than what is needed. To overcome these drawbacks, in this work the data was acquired using a Low Altitude Remote Sensing (LARS) approach with a small UAS. According to Anderson and Gaston (ANDERSON; GASTON, 2013), UAS can offers to ecologists a promising route to responsive, timely, and cost-effective monitoring of environmental phenomena at spatial and temporal resolutions that are appropriate to the scales of many ecologically relevant variables. Recently, Samiappan et al. (2016a) and Casagrande et al. (2017) presented a method for mapping regions with *Phragmites* based on true-color high-resolution imagery acquired with UAS. In the first work, the authors investigated the use of four texture analysis methods, GF, GLCM, SFTA, and WTA, with Support Vector Machine (SVM). Among the approaches in their investigation, they showed that the combination of GLCM with SVM provided the best result. Casagrande et al. (2017) showed that WTA with PNN presented improvements in several statistical variables such as overall accuracy and kappa value. These works can be considered the state-of-art in the scope here specified.

Considering that in this work was investigated the use of true-color high-resolution imagery acquired by a low altitude remote sensing platform, it is known that the computational methods used to map *Phragmites* play an essential role in the system accuracy. Recognizing this context, it was formulated the research question for this work: How to develop a new solution for mapping *Phragmites australis*, using low altitude remote sensing images acquired in the visible spectrum, through the use of texture analysis and machine learning techniques, capable of reducing the computational cost and spatial error?

1.1 OBJECTIVES

This section presents the general and specifics objectives of this work.

1.1.1 General

Develop an efficient framework for mapping *Phragmites australis* using low altitude remote sensing data acquired by a unmanned aerial systems through a classification system based on texture analysis and machine learning.

1.1.2 Specifics

- Improve mapping overall accuracy in true-color high-resolution imagery using binary classifiers and texture analysis techniques;
- Compare the mapping accuracies and computational efficiency of existing methods;
- Reduce the computational cost through a General-purpose Graphics Processing Unit (GPGPU) based framework implementation;

1.2 HYPHOTESIS

Use an ANN architecture with parameter optimization aiming to improve classification overall accuracy and implement the texture analysis and machine learning algorithms in the GPU aiming to reduce the computational cost.

1.3 JUSTIFICATION

The interest in ANN has been motivated by growing knowledge about the human brain (HAYKIN, 2003). This technique presents few advantages over other techniques, such as the large capacity for generalization (BENARDOS; VOSNIAKOS, 2007), less formal statistical training to develop, implicitly detect complex nonlinear relationships between

independent and dependent variables, ability to detect all possible interactions between predictor variables (TU, 1996). In addition, ANN approach has a distinct advantage over statistical classification methods, where it is non-parametric and require little or no a prior knowledge of the distribution model of input data (GANATRA et al., 2011).

As a consequence of all advantages, this technique is being widely used in a large number of applications, such as classification (THAKUR; MISHRA, 2017), pattern recognition (JALI et al., 2015), control systems (SRINIVASARAO; SREENIVASAN; SHARMA, 2016), among others. In remote sensing, ANN is being used mainly to improve the classification accuracy, such as (LIWA, 2006; YUAN; WIELE; KHORRAM, 2009). In (LIWA, 2006), the authors used an ANN to classify coastal marshes based on the phenological stages of plants. In (YUAN; WIELE; KHORRAM, 2009), the authors proposed an automated ANN system for land use/land cover classification.

In a direct comparison between SVM, which was used by (SAMIAPPAN et al., 2016a), and ANN, there is no consensus about which method will perform better for a specific problem. Some authors argue that SVM can present better results in a greater amount of problems since it is a very efficient and stable algorithm (AKANDE et al., 2014; SINGLA et al., 2011). In addition, these authors argue that ANN converge to local minima, considering that the optimization objective of ANN is multimodal in nature (AKANDE et al., 2014). Another argument that is commonly used against ANN is the difficult to settle its architecture a prior (HAYKIN, 2011), and, as a consequence, it tends to converge to local solutions too. As a counterpoint to these arguments, several authors have already proposed ways to solve or minimize the problems mentioned through optimization techniques (GANATRA et al., 2011; GARRO; VÁZQUEZ, 2015; BENARDOS; VOSNIAKOS, 2007). In addition, there are several works that presented comparisons between both classifiers in different data sets, where ANN presented a better result, as in (BELAKHDAR et al., 2016; PATEL; VALA; PANDYA, 2014; ANTKOWIAK, 2006). That is, both techniques have a great potential mainly in binary classification problems, but there is no clear pattern in which one of these two specific techniques will perform better. Considering that *Phragmites* maps reliability is essential for the resource managers, new solutions must be investigated, and among all techniques, ANN was chosen since it can offer the possibility of a better perform.

In addition to accuracy, the computational cost necessary to generate the map is an important factor to be considered. In (SAMIAPPAN et al., 2016a), the authors argued that the overall processing time per

square kilometer of their solution based on GLCM and SVM was approximately 50 min. Considering that this work will implement the same texture analysis methods described in (SAMIAPPAN et al., 2016a), the main difference is the classification algorithm. The optimization process, that is necessary to overcome the issues related to the ANN, is a process that has a high computational cost as well. Consequently, it is known that this solution based on optimized ANN probably will have a higher overall processing time per square kilometer considering a CPU implementation.

An alternative to reduce computational time, even using an artificial neural network, is to consider a GPU implementation of the proposed solution. According to (NVIDIA, 2017), CUDA is a parallel computing platform and programming model that harness the power of the GPU. As reported by (CHE et al., 2008), GPU can offer extensive resources, such as massive parallelism, high memory bandwidth, and general purpose instruction sets, including support for both single- and double-precision floating point arithmetic. As demonstrated by (CHE et al., 2008; JAROS; POSPICHAL, 2012) GPU implementation through CUDA obtained impressive speedups if compared to CPU implementation. In (CHE et al., 2008), the authors proposed the implementation of speckle reducing anisotropic diffusion and HotSpot, ANN Backpropagation, data encryption standard, Needleman–Wunsch global optimization, and k-means clustering. They described a speedup of 6 times in the ANN Backpropagation implementation if compared to GPU implementation. In (JAROS; POSPICHAL, 2012), the author proposed a fair comparison between CPU and GPU running the GA under Knapsack Benchmark and the GPU implementation achieved a speedup of 11,82 times. Furthermore, considering that the proposed solution split the original image into sub-images of 100 x 100 pixels and there is not a strict relation between these blocks, they can also be processed in parallel. Consequently, a CUDA implementation of the proposed methods is an alternative that will be investigated to reduce the necessary computational time to map *Phragmites*.

1.4 METHODOLOGY

Aiming to achieve the described objectives (Section 1.1), this work was developed by the following steps:

1. Do an exploratory bibliographical research in two main focus: Texture analysis and machine learning techniques, and solutions

to map *Phragmites* based on remote sensing;

2. Implement feature extraction and binary classifier methods chosen through step 1;
3. Optimize the parameters for the feature extraction methods. In this step will be used the same data described in step 4. Overall accuracy will be used for the parameter assessment;
4. Compare the maps generated from the proposed solution to the state-of-art, considering the kappa coefficient (κ);
5. Implement the proposed algorithms in the using CUDA aiming to reduce the computational cost;
6. Evaluate the GPU and CPU implementations considering the computational cost.

1.5 WORK STRUCTURE

This work is structured in 6 (six) chapters, including the introductory chapter (Chapter 1).

Chapter 2 presents the background knowledge necessary for this work. In addition, a resume about the main works published in the least fifteen years about methods for mapping of *Phragmites* using remote sensing is presented.

Chapter 3 describes the study area, the materials, and the methodology used in the image acquisition process. In addition, the ground reference map used in the evaluation step was presented.

Chapter 4 describes the pre-extraction and feature extraction steps used in this work. The four methods proposed are detailed in this chapter. In addition, the GPGPU implementation is discussed in details.

Chapter 5 will follow the same sequence described in chapter 3. However, instead of feature extraction algorithms, in this chapter will be defined all necessary steps to develop the binary classifiers techniques used in this work.

Chapter 6 will discuss and present the obtained results. First, a comparative study based on the spatial error between the techniques will be detailed. Then, the maps generated from the best combination will be presented. Finally, the computational cost is presented.

Finally, in chapter 7 this work will be concluded, highlighting the main contributions and opportunities for future work.

2 BACKGROUND

The process of data acquisition is an essential step for the solutions that are being proposed in this work. Low quality or non-standard data sets can drastically compromise the system accuracy. This fact can be explained by the system's dependence on the level of detail of the texture to be analyzed. That is, before the system itself, the resource manager must think about the equipment and methodology to be used for the data acquisition. Considering that, understanding the basic principles of remote sensing is as important as understanding the specificities of the solutions proposed here. In the following sections, the basic concepts needed to understand the solution and the image acquisition process are presented together with an analysis of the main related works.

2.1 REMOTE SENSING

There are two main categories of spatial data acquisition:

1. Ground-based methods: In-situ measurements, land surveying, field observation, among others is the operation in real world environment.

2. Remote sensing methods: Based on the data acquisition through sensors such as aerial cameras, radar or scanners. The information is derived from the data acquired through the sensors, which has a limited representation of the real world.

RS has been defined in many different ways by different authors, being the definition made by Campbell (1996) was adopted in this work. RS can be applied in many fields, including architecture, medicine, industrial quality control, robotics, etc. This work, however, is focused on the earth observation. As described in the introduction, although it has some negative aspects associated with the process of remote sensing, this methodology guarantees a good number of advantages. In some cases, the use of remote sensing is essential to make an application feasible. For example, although it is possible to map *Phragmites* using ground-based methods, it is not feasible mainly by the associated hazards and the total area to be covered.

2.1.1 Historic Context

Remote sensing of application started in 1859 by Gaspart Tournachon that took an oblique photograph of a small village near Paris from a balloon (AGGARWAL, 2003). He was responsible to start a trigger. After this first report of data acquisition through any platform, other people all over the world followed this example. New reports emerged from the Civil War and World War I. During these war periods, new technologies have emerged and developed remote sensing. Aircraft instead of balloons were tested during the World War I, for example. The military discovered that aircraft proved to be more reliable and more stable platform for earth observation than balloons (AGGARWAL, 2003).

As described by Aggarwal (2003) the civilian officially started to use remote sensing between the World War I and World War II. First applications of airborne photos in that time are geology, forestry, agriculture and cartography. These new applications helped to develop new technologies. New types of data acquisition spanned, such as near infrared, thermal infrared, and radar. After the wars, the color infrared emerged as well. In the 60s, data were already collected from space (NASA, 2017). In 1956, Colwell published the first reports of an important non-military use of color infra-red film and conducted experiments for the classification and recognition of vegetation type and the detection of diseases and damaged or stressed vegetation (COLWELL, 1956). From this point, remote sensing maintained a good level of development and use. Today, this method is increasingly being used for problem-solving. Refer to (AGGARWAL, 2003) for more details about remote sensing history.

2.1.2 Electromagnetic Energy

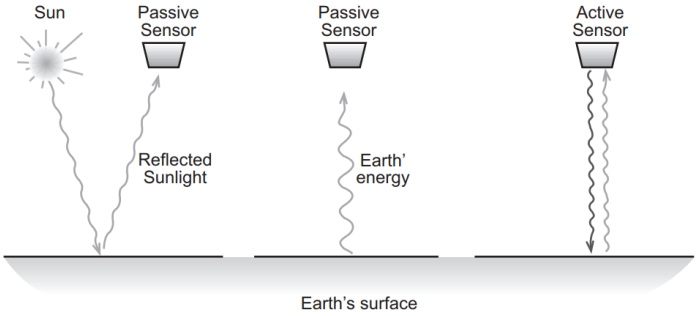
In most situations, RS relies on the measurement of electromagnetic energy. Unless it has a temperature of absolute zero ($-273°C$) an object reflects, absorbs, and emits electromagnetic radiation in a unique way, and at all times (NASA, 2017). This energy, that is emitted in waves, originates from vibrating electrons, atoms, and molecules. They absorb and emit electromagnetic radiation in unique combinations, called spectral signature. The entire array of electromagnetic waves comprises the electromagnetic spectrum.

Wavelength and frequency are two characteristics of electromag-

netic waves that are important for remote sensing. Wavelength is the distance between successive wave crests. Frequency is the number of cycles of a wave passing a fixed point over a specific period of time. Mostly the wavelength is usually used to describe specific places in the electromagnetic spectrum where the object emits more energy. The amount of energy radiated by an object depends on its absolute temperature, emissivity and is a function of the wavelength (JANSSEN; HUURNEMAN, 2001).

In RS, data acquisition can be active or passive . In the active methodology, the equipment has its own source of energy, such as radar and laser, where it emits its energy and measures the amount of energy reflected back. The passive technique uses natural source of energy. In the specific case that the source of energy is the sun, this technique can work only during daylight. This work uses only true-color imagery, consequently the sensor used is passive. Figure 1 shows a demonstration of the cited methodologies.

Figure 1 – Active and Passive techniques. The first has its own source of energy and can be used at any time. The second depends on external sources of energy, and in some cases, can be used just during daylight.



Source: Image from Janssen and Huurneman (2001).

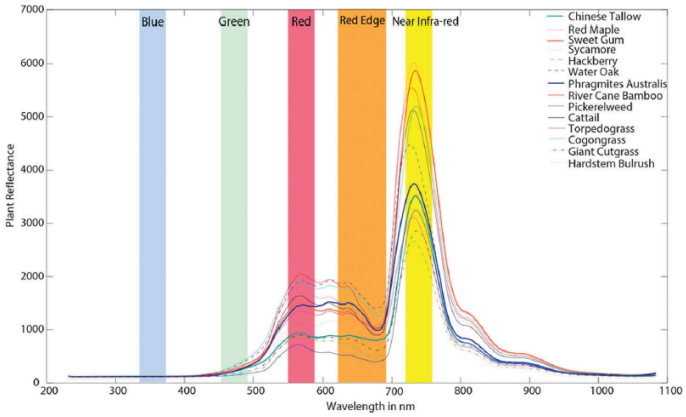
2.1.3 Energy Interaction with the Earth's Surface

In the scope of the application being proposed in this work, the reflected solar energy is the most important information since it helps to describe surface characteristics. In soil and water applications, part

of the energy is reflected, absorbed and even transmitted. Absorption occurs when radiation is absorbed by the object, whereas transmission occurs when radiation pass through the target. According to Janssen and Huurneman (2001), reflection can be further divided into two types:

- Specular reflection: occurs when the surface is smooth and almost all of the energy is directed away from the surface as a mirror-like reflection;
- Diffuse reflection: occurs when the surface is rough and the energy is reflected almost to all directions.

Figure 2 – Response of 14 different wetland plant species and bandwidth of each spectral band of sensor. As related by Samiappan et al. (2016b), the imagery has three visible bands blue (480 nm), green (560 nm), and red (670 nm), as well as red-edge (720 nm) and near-infrared (840 nm) bands. Despite the near spectral signatures, the difference between plants is perceptible.



Source: Image from Samiappan et al. (2016b).

In nature, it is common to find a mix of both types of reflections. The energy that reaches the surface is called irradiance and the reflected radiance. Each object has its own spectral signature. This spectral signature can be established through the reflectance of the object. In the application here proposed, this type of information can drastically modify the proposed solution, because the higher the spectral resolution

is, the lower the level of difficulty for the classification process. Figure 2 shows an example of different spectral signatures.

2.1.4 Digital Images

Images constitute a spatial distribution of the irradiance at a plane (JäHNE, 1997). A basic monochromatic image is usually a function of two spatial variables $f[x, y]$, which represents the brightness f at the Cartesian location $[x, y]$. This idea can be extended to three dimensions, with brightness displayed in the z-axis (EASTON, 2017). A real image, however, has a higher level of representation complexity, where time and spectral descriptors can also be used to represent the information being captured.

It is known that until the present moment computers cannot handle continuous images, but only arrays of digital images. Sampling and quantization are the two processes used to converter this continuous data captured through a sensor in a digital image, where quantization is the process of digitizing the average irradiance and sampling is the process of digitizing the values of the coordinates.

The quantization capacity is an important factor for the distinction of different irradiance levels. For example, a 16 bits irradiance value can be represented in 65536 different ways while an 8 bits value in only 256 ways. In solutions where classification is strongly related to the spectral signature of objects, the quantization capability is essential to better distinguish objects with close signatures. However, considering that this work is based on texture feature analysis, even more important than the quantization capacity is the sampling capacity.

The size of the region that the pixel represents, determines the level of detail of the data acquired. There is no specific rule to define how many points (pixels) an image must have to represent a data, but a general rule used is that, for a specific task, the pixel size should be smaller than the finest scale of the objects in study. The sampling capacity is an essential factor to define the size of the region, that is, the spatial resolution.

In remote sensing, a common way to define the spatial resolution of a sensor is called GSD. This measure is the distance, in field units, between the centers of two neighboring detectors, i.e. two neighboring pixels on the image (ORYCH, 2015). For example, in an image with a ten-centimeter GSD, adjacent pixels image locations are 10 centimeters

apart on the ground. The definition of this setting directly determines how sharp the final data will be, as demonstrated in Figure 3.

Figure 3 – Comparison between images with different GSD. (a) UAV imagery with GSD of 4cm, (b) Manned aircraft with GSD of 50cm.

Source: Images from authors.

In this work, GSD is defined according to 2.1 (PIX4D, 2017).

$$GSD = \frac{S_w * H * 100}{F_R * ImW} \tag{2.1}$$

Where H is the height at which the sensor is located relative to the object under analysis in meters, S_w is the sensor width in millimeters, F_R is the focal length in millimeters, and ImW the image width.

2.1.5 Sensors and Platforms

The measurements of this energy are made by sensors that are attached to remote platforms. Different types of sensors can be used for different types of applications. In addition, the platform and its specificities can modify important parameters that are essential for the development of the solution, such as final spatial resolution and quality of the data acquired. That is, the sensor-platform combination determines essential parameters of the resulting image data.

There is a growing community studying and discussing different approaches to map *Phragmites australis* efficiently. Accordingly to Mozdzer and Zieman (2010) *Phragmites* is one of the most widely researched plant species because of its perceived benefits and/or threats to ecosystem health and services. In addition, it often spans multiple years and multiple spatial scales, from small individual patches to whole landscapes (ADAM; MUTANGA; RUGEGE, 2009). Consequently, there is a wide range of possible solutions and already proposed works

aiming to map this specific type of vegetation using remote sensing.

One platform that has been widely used for *Phragmites* mapping is the satellite. Its higher coverage capacity, lower post-processing dependence, and speed to acquire the data are important features used to justify its use. In (GILMORE et al., 2008; LANTZ; WANG, 2013; PENGRA; JOHNSTON; LOVELAND, 2007; BOURGEAU-CHAVEZ et al., 2013; VILLA et al., 2013), different combinations of satellite-sensors were used aiming reduce the platform's drawbacks and achieve a solution with higher accuracy. Gilmore et al. (2008) suggested the use of the QuickBird Satellite with a five-band data (panchromatic, blue, green, red, and near IR) with 65cm/pixel of spatial resolution for the panchromatic band and 2.62m/pixel for the other bands at nadir. In addition, they opted for a laser-based sensor scanner for height identification of objects. Lantz and Wang (2013), hypothesized that the eight multispectral bands and the panchromatic band possessed by the Worldview-2 satellite would increase classification accuracy if compared to the traditional satellites with only four multispectral bands. This hypothesis is totally based on the spectral signature potential. In this case, the spatial resolution is 0.46m for the Panchromatic and 1.84m for the other eight multispectral bands at nadir. Pengra, Johnston and Loveland (2007) further explored this aspect of spectral signature and proposed the use of 30m resolution EO-1 Hyperion hyperspectral sensor to create a raster map of this kind of vegetation using a Spectral Correlation Mapper algorithm. Two important aspects to highlight in this work: small spatial and high spectral resolution. In Bourgeau-Chavez et al. (2013), the authors proposed a new combination of satellite-sensor based on multi-season Radar data. In this work besides the spatial and spectral context, the temporal context plays an essential role in the system.

Another platform that is commonly used to acquire data and map invasive plants is manned aircraft, as demonstrated in (BECKER, 2009; ARTIGAS; PECHMANN, 2010; VILLA et al., 2013). The data acquired through this platform generally has a higher spatial resolution than satellites. Becker (2009) and Artigas and Pechmann (2010), for example, proposed the use of high-resolution hyperspectral data collected through a light aircraft to classify *Phragmites*. Villa et al. (2013) proposed a more robust solution based on multitemporal and multisensor remote sensing datasets to monitor the conservation status and to assess the morphological complexity of *Phragmites australis*. In the first and second data acquisition, the authors proposed the use of airborne hyperspectral data. The first data has a spatial resolution of 1m and the second 2m. After, data was acquired using satellites (GeoEye and

Worldview 2). The first acquired 4 bands (blue, green, red, and near IR) with a spatial resolution of 1.7m and the second 8 bands (coastal, blue, green, yellow, red, red edge, near infrared 1, and near infrared 2) with a spatial resolution of 1.9m.

The recent development of UAVs allowed a new methodology of remote sensing called LARS. The interest in this approach is growing mainly due to its cost, safety, and very-high resolution data. Consequently, recently, several works have proposed new *Phragmites* mapping techniques based on this methodology. Considering that the spectral signature is essential to segment the objects, some authors proposed the use of sensors capable to acquire data out of the visible spectrum, as demonstrated in (SAMIAPPAN et al., 2016b; ZAMAN; JENSEN; MCKEE, 2011). Zaman, Jensen and McKee (2011) used four band data (visible and near infrared) with high-resolution (25cm/pixel) data to identify *Phragmites* locations. Samiappan et al. (2016b) proposed the use of a five band multispectral data (red, green, blue, red edge, and near infrared) with a digital surface model in 8cm/pixel data. Other authors consider that the high spatial resolution could be enough to propose solutions to map this vegetation, as demonstrated in (HUSSON; ECKE; REESE, 2016; SAMIAPPAN et al., 2016a; CASAGRANDE et al., 2017). In all cases, the authors used a spatial resolution of 5cm/pixel.

2.2 PATTERN RECOGNITION

Despite having a long history, Pattern Recognition (PR) began to evolve in the 1960s. Before it, most of the contributions to the area were proposed aiming theoretical advances in statistics. The advent of computers increased the demand for practical applications of PR. This field is concerned with the design and development of systems that recognize patterns in data. In a more formal definition, PR is described as the scientific discipline of machine learning whose goal is the classification of objects into a number of categories or classes (KPALMA; RONSI, 2007). Pattern, more specifically, is an entity, vaguely defined, that could be given a name. In other words, pattern can be described as an entity of interest that someone wants to identify or recognize. Texture in image, footprint, voice of an individual, and speech are examples of patterns.

According to Rosenfeld and Wechsler (2000), PR is one of the most important functionalities for intelligent behavior and it is displayed in both biological and artificial systems. Biological organism

must be capable to understand the data collected and respond appropriately for survival. In artificial systems, some form of sensing is used to identify specific characteristics. For example, a machine may be able to classify if a ceramic tile is within the norm using visual inspection. This classification would be based on features extracted from the data acquired through a sensor, where in this example the data type could be an image in the visible spectrum. This classification process might fail by some factors, and if this happens, this data can be saved for further analysis. This is an example of a pattern recognition application.

Although some authors disagree on the number of steps, the final process is well defined as demonstrated in (ROSENFELD; WECHSLER, 2000; DUTT; CHAUDHRY; KHAN, 2012; ELIE, 2013). PR approaches are typically composed of four steps: data acquisition and collection, preprocessing, feature extraction, and classification. The PR techniques in this work follow this description.

Data acquisition, that is the process of measuring some specific signal through a sensor, is an essential step for the final result. Poor quality and out of specification data can drastically compromise the result of the solution generated. The background knowledge needed to understand this work can be found in the previous section.

The pre-processing step is optional. However, this is a desirable step since operations at this stage may contribute considerably to the accuracy gain of the system. The most common operations in this step are: noise filtering and smoothing, data normalization to correct the data from different errors, and data segmentation to highlight the pattern from other objects.

Feature extraction methods aim to describe the relevant information contained in a pattern. This is a crucial step, since the the correct choice of feature is the key to system success. According to Elie (2013), good features must satisfy the three following requirements. First, Intraclass variance must be small. This means that features extracted from the same object must be close. Next, interclass separation should be large, which means that features extracted from different objects should have some value difference. Third, the dimensionality of the extracted feature. As a consequence of the complexity associated to a larger vector, the system's performance can decrease and the computational complexity became larger. In this case specifically, pre-processing operations can be applied aiming to reduce the impact of a complex data.

After this process, this extracted data must be assigned to a class. In the classification step, the classification approach attempts

to define a set of classes and labels for each input value. The number of possible classes represent the number of possibilities for the object in analysis; e.g., classify a set of images of fruits that can be oranges, apples, pears, or bananas. The number of labels represents the number of proprieties. For example, the shape of the fruit and the color.

PR is generally categorized according to the type of learning used to generate the output value (ELIE, 2013). There are two different categories: Supervised learning and unsupervised learning. In the first case, the methodology needs an training set that consists of a set of instances that have been labeled. Unsupervised learning, on the other hand, attempts to find inherent patterns in the data that could be used to correctly assign the correct class and label for new data instances.

As already described, several applications have already been proposed in order to solve the problem of mapping *Phragmites*. As presented in (BROOKS, 2014; LANTZ; WANG, 2013), maximum likelihood is a common classifier algorithm used, where it assumes that the statistics for each class in each band are normally distributed, calculating then the probability of a given area belongs to a specific class (SHAFRI; SUHAILI; MANSOR, 2007). Brooks (2014) proposed the use of this algorithm based on the spectral signature and majority filter to smooth the result's noise. She related that distortion, blurring, and inter-mixing of the vegetation were the main limitations of the proposed solution. Lantz and Wang (2013) reported a complete solution based on object-based and per-pixel maximum likelihood classification. The use of different rules for the same object using experimental values of threshold can be a problem for different applications. However, the use of a specific processing flow for shadowed objects increased the overall accuracy of the solution.

Another supervised learning algorithm commonly used in processes of *Phragmites* mapping is spectral angle mapper, as reported in (PENGRA; JOHNSTON; LOVELAND, 2007; BECKER, 2009; ARTIGAS; PECHMANN, 2010). This algorithm is a physically-based spectral classification that uses an n-dimensional angle to match pixels to reference spectra (SHAFRI; SUHAILI; MANSOR, 2007). In these three papers, the authors proposed the creation of spectral libraries aiming to identify different classes. After, a pixel-wise classification was performed through the spectral angle mapper. As shown in the works and considering the spectral angle mapper characteristics, this algorithm is usually chosen when the data has a high spectral resolution.

In addition to these alternatives, many authors propose the use of more than one pattern recognition flow in order to treat each object

in a specific way, as demonstrated in Bourgeau-Chavez et al. (2013), Husson, Ecke and Reese (2016). In Bourgeau-Chavez et al. (2013), the authors presented a solution based on iterative process based on isodata unsupervised classification. However, in mixed vegetation areas where *Phragmites* was heavily confused with other vegetation types, spectral signatures of identified classes from the isodata unsupervised classification algorithm were extracted and used for supervised, maximum likelihood classification. Husson, Ecke and Reese (2016) used three different techniques to generate the map: water versus vegetation using threshold classification, growth form using threshold classification and random forest, and dominant taxon classification through random forest. In the threshold classification, the thresholds used were determined empirically based on expert knowledge, limiting the solution's usage.

Other methods have already been proposed. In Zaman, Jensen and McKee (2011), the authors used a combination of high-resolution multi-spectral and temporal data in a multiclass relevance vector machine algorithm to produce quantitative land cover descriptions that identify *Phragmites* locations. Samiappan et al. (2016b), presented a new approach based on digital surface models, normalized difference vegetation index, soil-adjusted vegetation index, and morphological attribute profiles with SVM. In (SAMIAPPAN et al., 2016a), the authors described a comparison between four techniques of texture analysis with SVM. The tests conducted with GF, GLCM, SFTA, and WTA by the authors indicated that the texture-based approaches were suitable to map this kind of vegetation. The main difference between (SAMIAPPAN et al., 2016b) and Samiappan et al. (2016a) are the features used and the type of classification. In the first, the authors proposed a pixel-wise classification. In the second, a block classification. Block classification approach was proposed by Casagrande et al. (2017) as well, where the authors presented an improvement in the solution using WTA as texture analysis technique with PNN instead of SVM.

Although this work proposed the use of the same texture analysis techniques and PNN that were already reported in previous works, it differs by:

- Including a parameter tuning step in the execution flow;
- Using and comparig different neural network architectures as binary classifiers;
- Paralleling the solution on the GPU.

That is, despite having some steps equal to the works considered

as state of the art, this work proposes modifications in the execution flow. The modifications aim to produce results that are within the proposed objectives and confirm the hypothesis described.

2.3 GENERAL-PURPOSE GRAPHICS PROCESSING UNIT

Increasing the processor clock speed has been one of the main methods to improve the computational capabilities of devices for some time. However, as described by Simonite (2016), Moore's law is dying and other ways must be found to make computers more capable. Unfortunately, due to various limitations in the manufacturing process, the possibility of increasing the clock speed is getting smaller and smaller. As suggested by International Technology Roadmap for Semiconductors, silicon transistors can only keep shrinking for another five years (COURTLAND, 2016). Increasing parallelism rather than increasing clock rate has become the primary engine of processor performance growth. This trend is likely to continue (GARLAND et al., 2008).

Today the parallelism capacity of a CPU is much smaller than that of a GPU. Modern GPUs have been at the leading edge of the chip-level parallelism. Today a CPU can offer the possibility of a few cores optimized for sequential serial processing while a GPU has a massive parallel architecture consisting of thousands of smaller, efficient cores. This difference is a reflection of the objective of each component. GPU architectures evolved aiming to handle the needs of real-time computer graphics, that is a task with tremendous inherent parallelism (GARLAND et al., 2008).

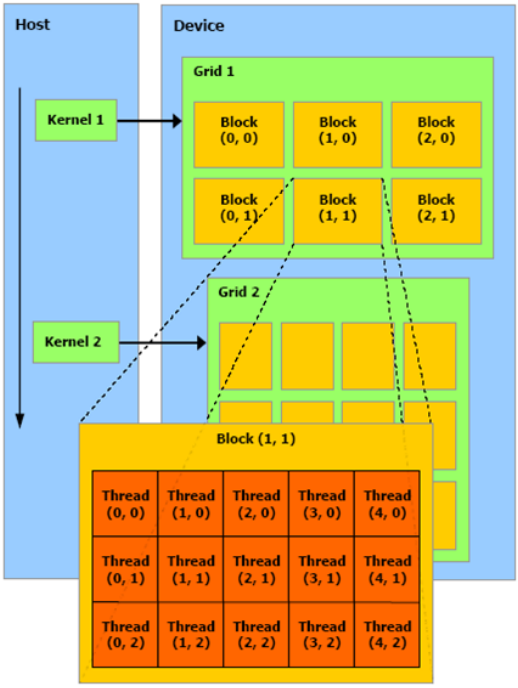
According to Riha and Smid (2011), there are three main technologies for general purpose programming on GPUs:

- CUDA by NVIDIA;
- BROOK+ from advanced Micro Devices (AMD);
- Open Computing Language (OpenCL), maintained by Khronos Group.

This work proposes the parallelization of the algorithms in the GPU through CUDA C extension. The main objective of this programming model is to use the GPU (device) as a coprocessor with its own Dynamic random access memory (DRAM) memory to the main CPU (host). NVIDIA describes this process as GPU-Accelerated computing, where just portions of the application are computed on the device,

that represents the GPU and the remainder of the code still runs on the host, that is CPU.

Figure 4 – Representation of a possible application with 2 kernels where the first was divided into 2x3 blocks with an internal size of 3x5 threads. That is, there are 6 blocks where each block has 15 threads.



Source: Image from Ding (2017).

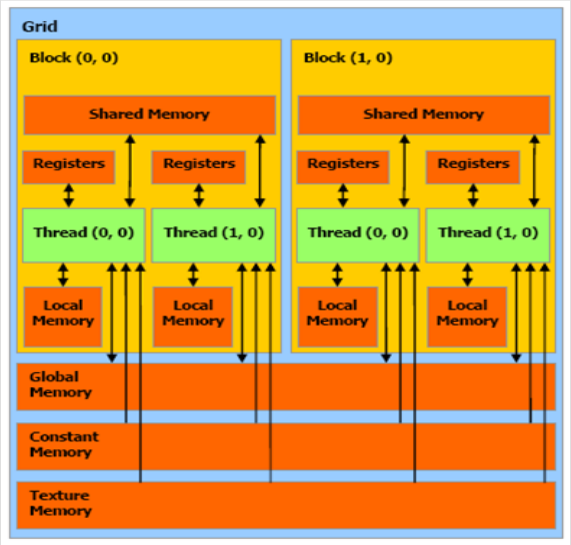
The host code manages the memory and the kernels that will be executed on the GPU, where kernels can be seen as a function which is executed on the device. However, prior to the GPU memory use, the kernel must allocate space and copy the data from host to device. After the use, this data must be copied again to the host. Afterwards, the host should also free the memory allocated on the device. The host is the responsible to launch and configure the kernels as well. The kernel is executed as a grid of thread blocks. A block is a batch of threads that can share data through a shared memory and that can be synchronized during the execution flow. Different blocks cannot access the same

shared memory and the threads cannot be synchronized. Figure 4 shows a possible application where the host started two kernels.

Although all threads and blocks execute the same code, as demonstrated in Figure 4, they have specific identifications. This creates the possibility of different behaviors when taking decisions, making computations and accessing the memory for each thread. In the kernel, the thread, block, and grid information are accessible via the following constants:

- threadID: Index of the thread in a block. This can be composed to up to 3 dimensions, where they can be addressed by: threadID.x, threadID.y, and threadID.z;
- blockID: Index of the block in the grid. It can be composed by one or two dimensions, where they can be addressed by: blockID.x and blockID.y;
- blockDim: Number of threads in the block;
- gridDim: Dimensions of the grid in blocks.

Figure 5 – CUDA device memory hierarchy representation.



Source: Image from Ding (2017).

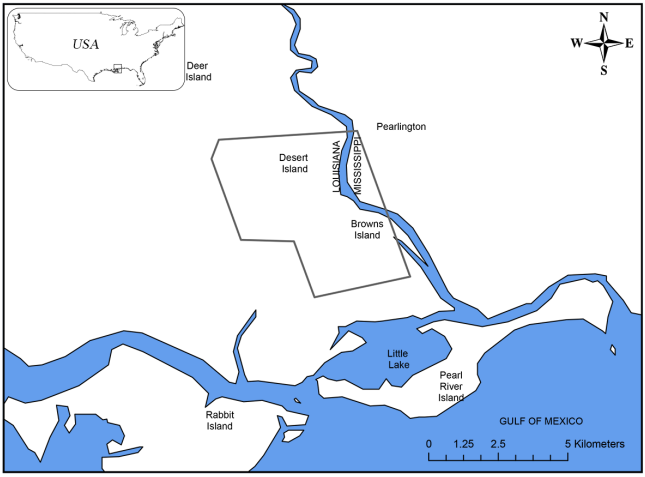
CUDA supports several types of memory that can be used by the developers. Each memory type has its specifics and must be correctly addressed aiming a high compute to global memory access ration. The possible types of memory in the GPU are: registers and local, global, constant, texture, and shared memory. For more details about the memory specifications refer to (RIHA; SMID, 2011). Figure 5 shows a diagram of CUDA Memory hierarchy.

Each memory type has its own specificities. Global, constant, and texture can be accessed by the host. Register and local can be accessed only by a specific thread. All memory types, except for texture and constant can be accessed for read/write operations. Constant and texture are read only. Register and shared memory are the unique on-chip. These characteristics must be taken into account when proposing a GPU implementation. A poor management can drastically reduce processing capacity of the system.

3 STUDY AREA

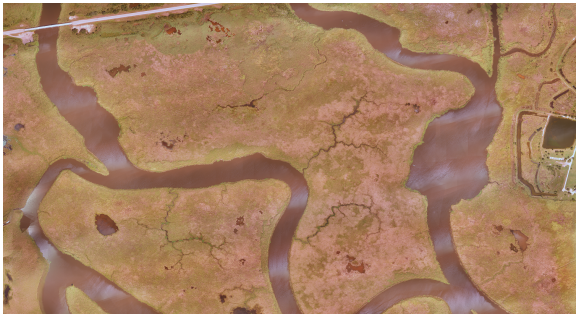
The study site near Pearlington, Mississippi, United States of America (USA) was selected to evaluate the proposed technique to map invasive *Phragmites* (Fig. 6). The region can be classified as a tidal freshwater marsh. Such regions are influenced by the daily influx of tides, yet they have a salinity of less than 0.5 ppt (CRONK; FENNESSY, 2001). This region, that is located in the delta of Pearl River and drains into the Gulf of Mexico, is one of the most intact river systems in the southeast USA (TEAM, 2001) with one of the healthiest marsh complexes in the USA (BIRD; KYLE, 2004). The Pearl River has been identified as a high priority focus for conservation within the Eastern Gulf Coastal Plain and Northern Gulf of Mexico eco-regions (BIRD; KYLE, 2004).

Figure 6 – The study site near Pearlington, Mississippi, USA (about 2600 acres) selected for developing and evaluating techniques to map invasive *Phragmites*.



Source: Image from Samiappan et al. (2016a).

Figure 7 – (I) Desert Island - total area of approximately 574ha



Source: Image from authors.

Figure 8 – (II) Browns Island - total area of approximately 489.35ha

Source: Image from authors.

Data was collected on 23 September 2014 in the lower Pearl River basin west of Pearlington, Mississippi north and south of US Highway 90. The analysis was conducted on data collected at two sites: (I) Desert Island (Figure 7) and (II) Browns Island (Figure 8), with a total area of approximately 2600 acres (1063 ha).

3.1 IMAGE ACQUISITION

The data was acquired using a waterproof Altavian Nova UAS that weighs approximately 7 kg with payload, with a 2.7 m wingspan and 1.5 m length (see Figure 9), and is capable of water landings. This system can capture data on flights lasting 90 minutes.

The camera used to acquire the images is a modified Canon Rebel EOS SL1 that captures true color images. The size of the images acquired is 5184×3456 pixels with 8 bits per RGBA channel. Considering the camera's specifications and an altitude of approximately 231m, the GSD maintained was approximately 5 cm/pixel side. Aiming to keep a pixel overlap in five images or more in the region of analysis, the flight plan was defined considering 50% side overlap and 70% forward overlap.

Figure 9 – Waterproof Altavian Nova UAS.

Source: Image from Altavian (2017).

The software used to create the mosaic was Agisoft Photoscan Pro. The latitude, longitude, and altitude that the UAS stored during

the flights were used to define initial camera positions in the software. Orthomosaic tiles were exported at a size of 3184 × 3184 pixels and stitched together into a virtual mosaic using Geospatial Data Abstraction Layer software. The geo-referencing was performed using only data obtained through the flight telemetry data. Ground control points and post-flight corrections were not used. The mosaics produced were within meters or better of their true position.

The image was white-balanced in a post-processing step. This process was necessary to remove unrealistic color casts in pixels that are actually white (MALONEY; WANDELL, 1986). The open source software RawTherapee was used to create white-balance profiles that were applied to the data.

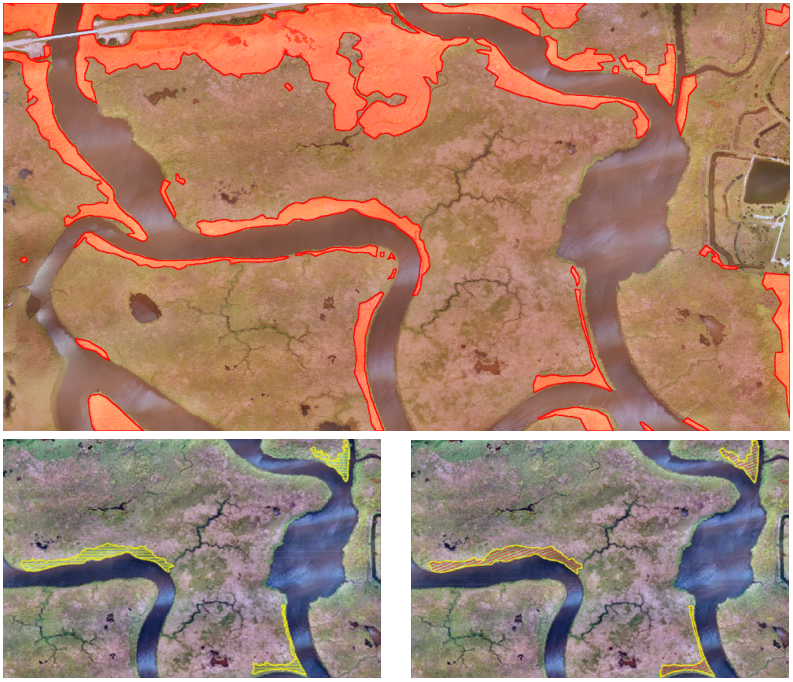
3.1.1 Ground Reference Map

To build the Ground Reference Map (GRM), in the work by Samiappan et al. (2016a), domain experts recorded the boundaries of three selected patches by walking around them with a Trimble Geo 7X GPS unit, with sub-decimeter accuracy. Navigation to and around patches was difficult due to location, terrain, and navigation. This reinforces the need to develop solutions based on remote sensing.

After collecting coordinates of GR patches, the true color image was loaded into Environmental Systems Research Institute ArcMap program. The boundaries of the same three patches were manually digitized by a domain expert based on direct visual inspection of the image mosaic in ArcMap. These digitized boundaries were then compared to the in situ patch GR boundaries. Samiappan et al. (2016a) discussed that the difference between both boundaries is almost insignificant (See Figure 10).

Aiming to physically verify the rest of the GR *Phragmites* patch locations along river channels and roadways, the authors of this work returned to the field. The authors navigate around these patches with a GPS unit or visit patches that were inland from a river channel or roadway during the revisit. These series of actions were performed with the intention of ensuring that the digitized boundaries could be used as a ground reference to verify the system's accuracy.

Figure 10 – Ground reference for study area I (*Phragmites* locations are outlined in red) (top). Ground reference patches of *Phragmites* determined by walking along the patch boundary and determining vertices using a handheld GPS unit (Bottom-left). Digitized boundaries – DIG from visual inspection (Bottom-right).



Source: Images from Samiappan et al. (2016a).

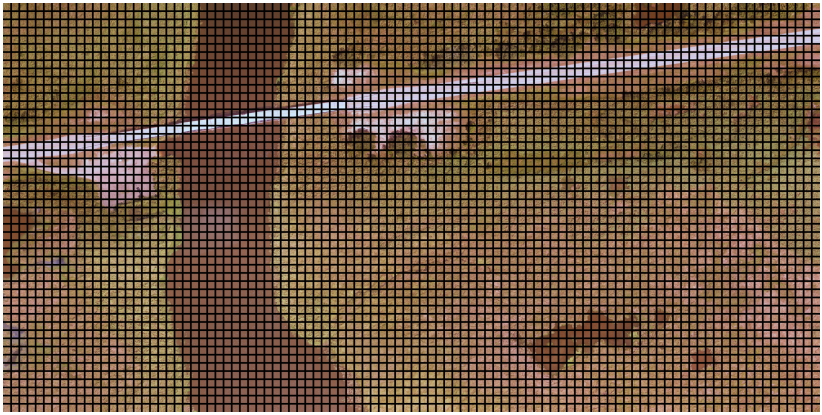
4 TEXTURE FEATURE ANALYSIS

The input to all proposed approaches consists of a true-color image in GeoTIFF format, that provides extra georeferencing information. Prior to any texture feature extraction method, the input image passes through three different steps that aim to adjust the data for the next stages.

First, the geoinformation that composes the GeoTIFF image is saved in a structure to be used in the post-processing stage. This information is essential and cannot be lost since the resource managers need to know the exact location of each *Phragmites* stand. In the post-processing step, this saved data will be reused in the map generation.

After, the image is divided into blocks of nxn pixels. This operation was performed aiming to reduce the computational cost and include the spatial context in the classification step. Considering that *Phragmites* generally creates mono and dense stands and that each pixel has only three spectral values, the spatial context is essential for the proposed solutions. In our study, one pixel of the study images represents a square area of approximately $25cm^2$. The optimal size for the blocks was derived via empirical analysis with various sizes, and it was defined as 100 x 100 pixels representing $25m^2$. A representation of the input image divided into blocks is shown in Figure 11.

Figure 11 – Segment from original set I represented through blocks.



Source: Image from authors.

Finally, these blocks need to be converted to grayscale by computing the luminance (Equation 4.1). This process is necessary because all texture feature extraction algorithms used in this work are limited to only 2 dimensions and the execution of each visible spectrum separately would increase considerably the computational cost without a gain that worth it. The conversion was based on the ITU-R BT601-7 recommendation.

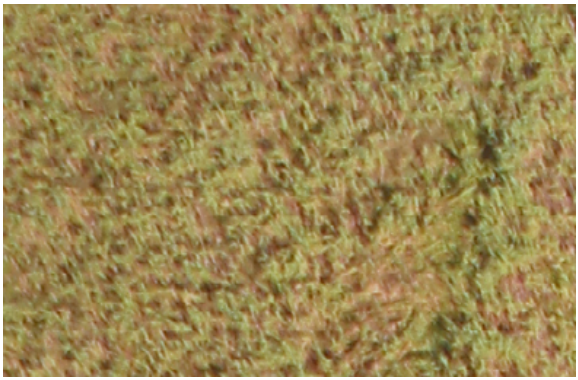
$$Y = 0.2989 * R + 0.5870 * G + 0.1140 * B \tag{4.1}$$

Where R, G, and B represents red, green, and blue channels, respectively.

4.1 FEATURE EXTRACTION

Feature extraction is an important process to compute characteristics of a digital image. The features should be able to describe a region and concisely quantify its texture content. This descriptor plays an essential role in image processing because many properties can be measured by it. Considering that in this work only the visible spectrum was used, features extracted from texture were used to compensate the lack of richness in the data collected.

Figure 12 – Segment from Site I representing *Phragmites* object.



Source: Image from authors.

A visual inspection of these imagery revealed unique properties

of *Phragmites*; mainly roughness, granulation, and regularity (Figure 12). Consequently, it was observed that the use of texture features is a solution for the distinction between *Phragmites* and other objects, as demonstrated in (SAMIAPPAN et al., 2016a). In this work was used the same four extraction feature algorithms proposed by (SAMIAPPAN et al., 2016a), being them: (1) GLCM, (2) Texture Features Based on GF, (3) SFTA, and (4) WTA.

4.1.1 Grey Level Co-occurrence Matrix

Initially proposed in (HARALICK; SHANMUGAM; DINSTEIN, 1973), GLCM is a square matrix, whose elements can reveal the relative frequency of occurrence of pairs of gray level values separated by certain distance or offset $(\Delta x, \Delta y)$. Δx is the number of columns between the pixel and its neighbors and Δy corresponds to the number of rows. The offset is generally expressed in polar coordinates as an angle Θ and distance ρ. If an intensity image were entirely flat, i.e. without texture, the resulting GLCM would be completely diagonal. As the image texture increases, the off-diagonal values become larger (BHARATI; LIU; MACGREGOR, 2004).

The GLCMs are normalized and stored in a I x I x N matrix, where N is the number of GLCM calculated due to the different displacements and orientation used in the computation. In this case, considering the computational time required to calculate each GLCM, only one value for displacement and orientation was used. In our experiments, the number of gray levels (I) was defined as 8, 135° for Θ, and 4 as distance ρ.

The co-occurrence probabilities provide a second-order method for generating texture features (HARALICK; SHANMUGAM; DINSTEIN, 1973). These probabilities represent the conditional joint probabilities of all pair wise combinations of gray levels in the spatial region of ρ and Θ. This co-occurrence probability between gray levels i and j is denoted as p_{ij} and is given by Equation 4.2.

$$p_{ij} = \frac{P_{ij}}{R} \tag{4.2}$$

Where P_{ij} represents the number of co-occurrences of i and j within the spatial region and R the number of cells in the matrix. The GLCM is then used for calculation of features that describe the texture of the image. In this work, besides the 14 characteristics proposed by (HAR-

ALICK; SHANMUGAM; DINSTEIN, 1973), it was used other 8 features proposed by (SOH; TSATSOULIS, 1999; GOMEZ; PEREIRA; INFANTOSI, 2012). The features can be described by the following equations:

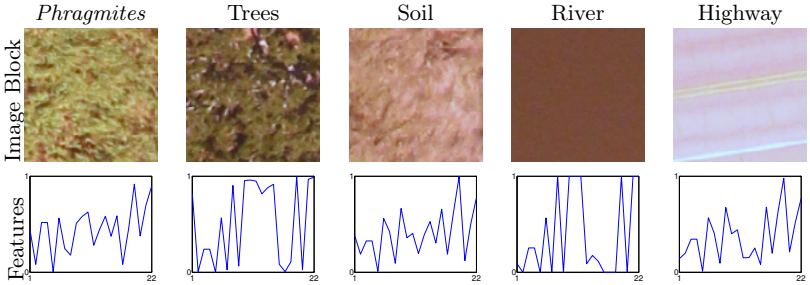
Table 1 – GLCM Features

Feature	**Formula**
Angular Second Moment	$f_1 = \sum_i \sum_j p(i,j)^2$
Entropy	$f_2 = -\sum_i \sum_j p(i,j) * log(p(i,j))$
Dissimilarity	$f_3 = sum_i \sum_j p(i,j) * \lvert i-j \rvert$
Contrast	$f_4 = \sum_i \sum_j p(i,j) * (i-j)^2$
Inverse difference	$f_5 = \sum_i \sum_j \frac{p(i,j)}{1+\lvert i-j \rvert}$
Correlation	$f_6 = \sum_i \sum_j \frac{i*j*p(i,j)-\mu_x*\mu_y}{\sigma_x*\sigma_y}$
Homogeneity	$f_7 = \sum_i \sum_j \frac{p(i,j)}{1+(i-j)^2}$
Autocorrelation	$f_8 = \sum_i \sum_j i*j*p(i,j)$
Cluster Shade	$f_9 = \sum_i \sum_j (i+j-\mu_x-\mu_y)^3 * p(i,j)$
Cluster Prominence	$f_{10} = \sum_i \sum_j (i+j-\mu_x-\mu_y)^4 * p(i,j)$
Maximum probability	$f_{11} = \underset{i,j}{MAX} p_{i,j}$
Sum of Squares	$f_{12} = \sum_i \sum_j (i-\mu)^2 * p(i,j)$
Sum Average	$f_{13} = \sum_{i=2}^{2\sum_i \sum_j \frac{p(i,j)}{1+(i-j)^2}} i * p_{x+y}(i)$
Sum Variance	$f_{14} = \sum_{i=2}^{2I} (i-f_8)^2 * p_{x+y}(i)$
Sum Entropy	$f_{15} = \sum_{i=2}^{2I} p_{x+y}(i) * log(p_{x+y}(i))$
Difference variance	$f_{16} = variance\ of\ p_{x-y}$
Difference entropy	$f_{17} = \sum_{i=0}^{I-1} p_{x-y}(i) * log(p_{x-y}(i))$
Information measures of correlation - 1	$f_{18} = \frac{HXY-HXY1}{max\{HX,HY\}}$
Information measures of correlation - 2	$f_{19} = (1-e^{[-2*(HXY2-HXY)]})^{\frac{1}{2}}$
Maximal correlation coefficient	$f_{20} = (second\ largest\ eigenvalue\ of\ Q)^{\frac{1}{2}}$
Inverse difference normalized	$f_{21} = \sum_i \sum_j \frac{p(i,j)}{1+\lvert i-j \rvert^2 / I^2}$
Inverse difference moment normalized	$f_{22} = \sum_i \sum_j \frac{p(i,j)}{1+(i-j)^2 / I^2}$

Source: Adapted from Haralick, Shanmugam and Dinstein (1973), Soh and Tsatsoulis (1999), and Gomez, Pereira and Infantosi (2012).

Where i and j are the index of the matrix, HXY is the entropy, HX and HY are entropy of $P_x(i)$ and $P_y(j)$, and the other variables are described in the Table 2.

Figure 13 – Image blocks examples (top), representing the five most common object categories in the study area, and their respective GLCM signatures (bottom) concerning 22 statistical properties.



Source: Images from authors.

Table 2 – GLCM Variables

Variables
$p_y(j) = \sum_{j=1}^{I} P(i,j)$
$p_x(j) = \sum_{i=1}^{I} P(i,j)$
$\mu = \sum_i \sum_j P(i,j)/I^2$
$\mu_x = \sum_i \sum_j ip(i,j)$
$\mu_y = \sum_i \sum_j jp(i,j)$
$\sigma_x = \sum_i \sum_j (i-\mu_x)^2 p(i,j)$
$\sigma_y = \sum_i \sum_j (i-\mu_y)^2 p(i,j)$
$HXY = -\sum_i \sum_j p(i,j) \cdot log(p(i,j))$
$HXY1 = -\sum_i \sum_j p(i,j) \cdot log(p_x(i)p_y(j))$
$HXY2 = -\sum_i \sum_j p_x(i)p_y(j) \cdot log(p_x(i)p_y(j))$
$p_{x+y}(k) = \sum_{i=1}^{N_g} \sum_{j=1}^{N_g} p(i,j),\ i+j=k\ e\ k=2,3,...,2N_g.$
$p_{x-y}(k) = \sum_{i=1}^{N_g} \sum_{j=1}^{N_g} p(i,j),\ \lvert i-j\rvert = k\ e\ k=0,1,...,N_g-1.$
$Q(i,j) = \sum_k \frac{p(i,k)*p(j,k)}{p_x(i)*p_j(k)}$

Source: Adapted from Haralick, Shanmugam and Dinstein (1973), Soh and Tsatsoulis (1999), and Gomez, Pereira and Infantosi (2012).

As described, the large mosaic was divided into sub-images of *nxn* pixels. However, in this case, GLCM features were extracted for each sub-image with an overlap of 20% between sub-images to avoid

the edge effect. Figure 13 (bottom) shows the 22 parameters calculated from GLCM in the most common images in both sites. From the data shown in the graphs, it is possible to determine that, in a specific comparison between the five graphs, the closest to the features extracted for *Phragmites* was the soil. However, there is a considerable difference between all blocks. For more details about GLCM algorithm implemented in this work, refer to (HARALICK; SHANMUGAM; DINSTEIN, 1973; MARCEAU et al., 1990).

4.1.2 Gabor Filters

GF were initially described in (GABOR, 1946) and later extended to 2-D in (DAUGMAN, 1985). It has been recognized as a very useful tool in image processing and recognition, and computer vision, mainly for texture analysis due to its capacity to acquire localization properties in both spatial and frequency domains. The most important advantage of Gabor filters technique is its invariance to illumination, rotation, scale, and translation. Furthermore, they can withstand photometric disturbances, such as illumination changes and image noise (HAGHIGHAT; ZONOUZ; ABDEL-MOTTALEB, 2013). In the spatial domain, a two-dimensional Gabor filter is a Gaussian kernel function modulated by a complex sinusoidal plane wave, and it can be defined in equation 4.3.

$$G(x,y) = \frac{f^2}{\pi\gamma\eta} exp\left(-\frac{x'^2 + \gamma^2 y'^2}{2\sigma^2}\right) exp(j2\pi f x' + \phi) \tag{4.3}$$

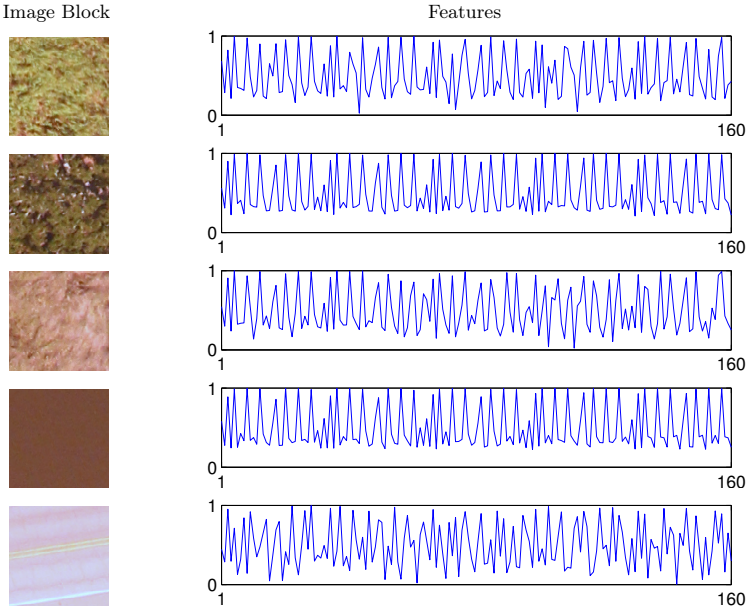
Where f is the frequency of the sinusoidal factor, Θ represents the orientation of the normal to the parallel stripes of a Gabor function, ϕ is the phase offset, σ is the standard deviation of the Gaussian envelope, γ is the spatial aspect ratio which specifies the ellipticity of the Gabor function's support, and x' and y' are defined as follows:

$$x' = xcos\theta + ysin\theta \tag{4.4}$$

$$y' = -xsin\theta + ycos\theta \tag{4.5}$$

As shown in the Figure 14, this texture analysis technique has a good ability to differentiate *Phragmites*, trees, and soil from the objects river and highway. However, the signature from the first three objects is

Figure 14 – Features from image blocks representing the five most common objects. From top to down: *Phragmites*, trees, soil, river, and highway.



Source: Images from authors.

relatively similar and this can confuse the classifier algorithm. Despite it, it is still possible to observe some specific points of change between the mentioned signatures.

We experimentally defined the number of orientations as 8 and scales as 5. Considering that the adjacent pixels in the image are highly correlated, it is possible to reduce the redundancy by downsampling the features images resulting from Gabor filters (SHEN; BAI; FAIRHURST, 2007; DUNN; HIGGINS, 1995). Since the column and row downsampling factors were defined as 50, the final feature vector was composed of 160 features (100 x 100 x 5 x 8) / (50 x 50). Examples of features extracted using this algorithm of the most common blocks are presented in Figure 14. For more details about feature extraction in Gabor Filters, refer to the work of (HAGHIGHAT; ZONOUZ; ABDEL-MOTTALEB, 2013).

4.1.3 Segmentation-based Fractal Texture Analysis

Filter functions such as Gabor is fairly useful in capturing texture information in the images. However, this techniques are very time-consuming because they need to consider multiple orientations and scales. In this type of work, where a large amount of high-resolution data is used, the computational cost that an algorithm requires is very critical. Considering this problem, fractal features were considered.

SFTA consists of two main methods. In the first method, the grayscale input image is decomposed into n_t binary images trough the Two-threshold Binary Decomposition (TTBD) algorithm (COSTA; HUMPIRE-MAMANI; TRAINA, 2012). After, the mean gray level and the number of pixel is computed from each binary image. In addition, the fractal dimensions are computed.

In the TTBD algorithm, the first step is to define the superior and inferior limit for each interval. Threshold pairs (T) that minimize the input image intra-class variance are computed by recursively applying a multilevel Otsu algorithm (LIAO; CHEN; CHUNG, 2001) n_t times. The number of times is a parameter defined by the user and has an influence on the feature vector size. The binary image for each threshold is obtained applying the original image in the equation described in 4.6.

$$L_b(x,y) = \begin{cases} 1, \; if \; t_l < L(x,y) \leq t_u \\ 0, \; otherwise \end{cases} \tag{4.6}$$

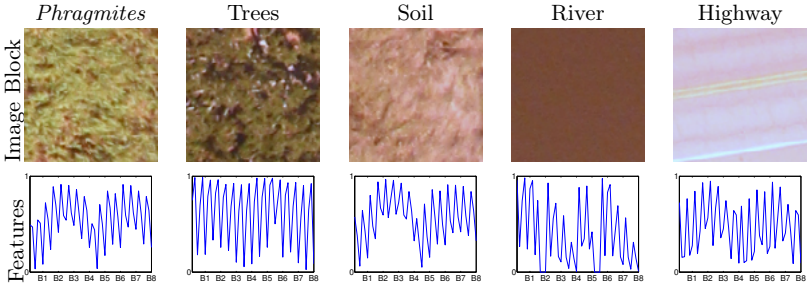
Where $L(x,y)$ is the input image, t_l and t_u are the lower and upper threshold values and $L_b(x,y)$ is the binary image.

The set of binary images is obtained applying equation 4.6 on the input image using all pairs thresholds of contiguous thresholds from $T \cup n_l$ and all pairs of thresholds $t, n_l \in T$, where n_l corresponds to the maximum possible gray level in the input image. As a consequence of this step, the number of resulting binary images is $2n_t$.

After applying TTBD to the grayscale image, the feature vector is defined by the pixel quantity, mean gray level, and fractal dimensions from each binary image. Fractal dimensions are described by the complexity of the object's boundary and are represented as a border image $L_e(x,y)$. This value is obtained through the equation 4.7.

$$L_e(x,y) = \begin{cases} 1, if \exists (x^{'}, y^{'}) \in N_8[(x,y)], \\ L_b(x^{'}, y^{'}) = 0 \wedge L_b(x,y) = 1, \\ 0, otherwise \end{cases} \tag{4.7}$$

Figure 15 – Image blocks examples (top), representing the five most common object categories in the study area, and their respective SFTA features (bottom) calculated using $n_t = 8$.



Source: Image from authors.

Where $N_8[(x, y)]$ represents the region of pixels that are connected in eight possible directions to (x, y). $L_e(x, y)$ gets value 1 if the (x, y) in the corresponding $L_b(x, y)$ has a value of 1 and $(\wedge)$ having a minimum of one neighboring pixel with value 0. The region boundary set gets a value of 0 otherwise. The fractal dimension is computed from $L_e(x, y)$ by using a box counting algorithm described in the work of (JR et al., 2010).

The second and third feature represent the mean and size (pixel count) of the binary images, respectively, and are calculated using the binary images generated by TTBD. These features bring the advantage of complement the extracted information without increase too much the computational time. The final size of the feature vector is three times the quantity of binary images produced by TTBD.

Figure 15 (bottom) shows these parameters for the most common blocks in both sites. From Figure 15, it is possible to conclude that the most similar object to *Phragmites* is the soil. However, it is visible the difference between the five objects. In this work, the final n_t was defined experimentally as 8. For more details about the SFTA method, refer to the work of (COSTA; HUMPIRE-MAMANI; TRAINA, 2012).

4.1.4 Wavelet Texture Analysis

The human visual system is very effective at interpreting spatial frequency of the luminance channel at multiscale. This ability

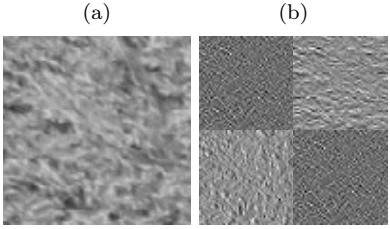
can be emulated by means of wavelet transforms. Wavelet transforms enable multi-resolution analysis of features by decomposing data $X = \{x_1, x_2, ..., x_N\}$ into two sub-bands $L|H = \{l_1, l_2, ..., l_{\frac{N}{2}}, h_1, h_2, ..., h_{\frac{N}{2}}\}$, where L represents the low-pass component and H represent its high-pass counterpart. This can be extended to bi-dimensional data, where four sub-bands are obtained by performing low- and high-pass decomposition in both horizontal and vertical directions. In this work, each grayscale block is applied to the Haar wavelet transform (STANKOVIĆ; FALKOWSKI, 2003), that extracts the low-pass components by averaging and the high-pass counterpart by differencing, according to the following equations:

$$l_i = \frac{x_{2i} + x_{2i+1}}{2} \tag{4.8}$$

$$h_i = \frac{x_{2i} - x_{2i+1}}{2} \tag{4.9}$$

Where x_{2i} and x_{2i+1} are row transformed pixels.

Figure 16 – One level of Haar wavelet transform applied to (a) example block with 100×100 pixels. The output (b) is divided into four sub-regions with 50×50 pixels each. Top-left (LL) shows the approximation values, top-right (LH) shows details in horizontal, bottom-left (HL) shows details in vertical, and bottom-right (HH) shows details in diagonal.



Source: Images from authors.

A transformation given by Equations 4.8 and 4.9 is employed horizontally to the image block, dividing it into two sub-regions concerning the two sub-bands L (low-frequency) and H (high-frequency). Later, these sub-bands are used as input to a second pass, now vertically, which results in four sub-bands with 2:1 downsampling. Fig. 16b shows the four sub-bands (i.e., approximation LL, details in horizon-

tal point LH, details in vertical point HL, and details in diameter HH at top-left, top-right, bottom-left, and bottom-right, respectively) obtained after applying one level Haar wavelet transform to Fig. 16a. The input images were captured at an approximate constant and low ground distance by the UAS, thus, more than one level Haar transform is not required.

In order to represent spatial frequency, from each sub-band, it was computed four statistical parameters, i.e., mean, standard deviation, entropy, and energy value, composing 16 parameters of each image block. Entropy is often described as a measure of randomness and energy a measure of the frequency distribution. These values can be computed as described in the Table 3.

Table 3 – Wavelet Features

Mean value	$\mu = \frac{1}{N}\sum_i p_i$
Standard deviation	$\sigma = \sqrt{\frac{1}{N}\sum_i (p_i - \mu)^2}$
Entropy	$S = -\sum_i p_i \log p_i$
Energy	$E = \sum_i \lvert p_i \rvert^2$

Source: From authors.

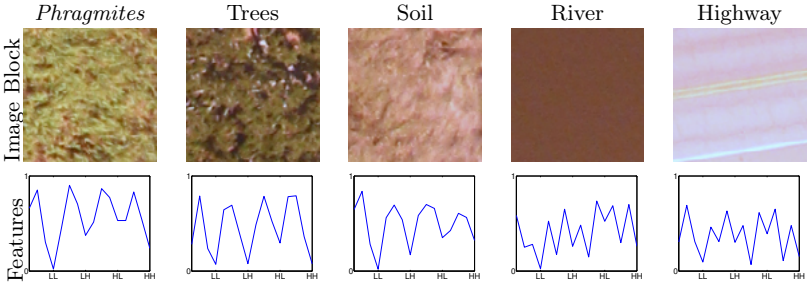
Where p_i are pixels' values and N is the total number of pixels.

Figure 17 (bottom) shows the values of these four parameters for the example blocks on the top row. Note the difference in the values of parameters for each block category, which makes these parameters suitable for classification of *Phragmites* in this context. Refer to the work of (LIVENS et al., 1997) for more details about the WTA algorithm implemented in this work.

4.1.5 GPU Implementation

The described feature extraction step can be easily structured in a GPU implementation mainly by the way the solutions were structured. The block processing purpose created a good opportunity for parallelization, because these blocks does not have any time of relation. That is, these blocks do not share the same memory, do not rely on data from the other blocks, do not need to be synchronized, and among

Figure 17 – Image blocks examples (top), representing the five most common object categories in the study area, and their respective wavelet texture signatures (bottom) concerning four statistical properties (mean, entropy, standard deviation, and energy value) for each sub-band (LL, LH, HL, and HH).



Source: Images from authors.

others.

It is known that different approaches could be used to reduce the necessary computational time through the use of a GPU. One of these approaches is to consider each block as an independent process. In this case, the feature extraction algorithm would be processed in linear, but the blocks no. Another possibility is to consider the parallel implementation of the method, where each block would be processed in serial but the feature extraction method would be processed in parallel. Examples of GPU implementations of the methods discussed in this chapter can be found in (WANG; SHI, 2010; FRANCO et al., 2009; HONG; ZHENG; PAN, 2017), where in the first the authors proposed a Fast Gabor Filters GPU implementation, the second 2D Wavelet transform, and the third GLCM. A third feasible option would be the use a two levels of parallelization, where the block process would start a new kernel to process the data in parallel. In this work, considering that a parallel implementation of each feature extraction algorithm would cause big changes in the structure, the first option was chosen to be tested.

Each image block is processed as follows:

- Image is read into host memory;
- Data is copied to the device and saved as texture;

- Grid is defined for the RGB to Grayscale step. Each thread was responsible for one pixel and the blocks were responsible for a group of threads.
- RGB to Grayscale parallel operation is performed and the data is saved in the global memory;
- A new grid is defined for the feature extraction process. Each is represented by a clock;
- The feature extraction process happen here;
- The resulting data is copied back from the global memory to the host.

In this scenario, two different kernels were identified: RGB to gray scaçe conversion and the feature extraction process. In the first one, small blocks of 10 x 10 threads were used. The number of blocks is a consequence of the image size and the number of threads per blocks. In this case, each pixel was treated by one thread.

Threads were not used in the second process. Each grid block represented a real image block. The grid block size depends on the total image size. For example, for Site I the final grid block size was defined as 335 x 190, where the total image size is 33500 x 19000 pixels. After identigying the position of the block in the grid, the linear process of each extraction method is normally performed.

5 CLASSIFICATION SYSTEM

After the feature extraction step, a class must be assigned to each block. For this task, three different ANN architectures were compared, being PNN, RBFN, and MLP. In addition, SVM, which is the classification algorithm used in Samiappan et al. (2016a) is also discussed in this chapter. However, prior to the class assignment, pre-classification and hyperparameter tuning steps must be performed in order to better adjust the data and parameters used in the algorithms.

5.1 PRE-CLASSIFICATION

The classification system receives from the texture feature analysis an array that is composed by $N \times F$ values, where N is the number of blocks and F is the number of features per block. These values are then linearly scaled to values in the range $[0, 1]$. This process is necessary to normalize the numerical difference between the extracted features. As reported by Sola and Sevilla (SOLA; SEVILLA, 1997), this normalization can have two side effects. First, the network performance is enhanced when input is equalized. Second, after a certain number of iterations and with some normalization methods, the square mean error tends to rise again. The first effect can be explained by some facts: in the MLP and RBFN architectures the weights are randomly defined in the (-1,1) interval, Hyperbolic tangent and Log-sigmoid transfer function, with unitary slope, are being used in the MLP algorithm, reduction of the distance to be covered by the backpropagation algorithm, specific features will not have a greater importance in the error reduction algorithm due to an uneven scale, among others. The counterpoint from this process is explained by the proximity of the data, i.e. when the training data are fitted more closely, its performance for other data start to become worse.

In the final phase of the pre-classification step, it is necessary to define the cases that will be used in the hyperparameter tuning step. For such, it is used a ground reference map that indicates the regions with *Phragmites* in the input map. 20% from the whole input map area was used to select the training set. From this area, only 10% was used in the training step. The remaining blocks, which are not known by the classifier, were used for the system evaluation.

However, prior to the hyperparameter tuning step of each com-

bination of texture-classifier, a search operation is performed aiming to find a viable training set. This process is performed in order to find a set that could represent the maximum possible area. For example, using a random selection of training cases could result in a training set selected from an area with specific illumination. Another possibility is a specific object that is not represented in the training cases. In both examples, the system final accuracy could be considerably reduced by a training set selection error.

In this training set search operation, the blocks were randomly selected. In the first set, the total number of blocks that are completely inside regions with *Phragmites* could range from 20 to 52% of the total number of *Phragmites* blocks in the Site. In the second set, the total number ranged from 27 to 62%. The remaining training set percentage was completed with not *Phragmites* cases. The other 10% not randomly selected blocks were used to check the training set fitness. The Mean Square Error (MSE), that is described in the Equation 5.6, is calculated from the classifier's outputs.

5.2 SUPORT VECTOR MACHINE

SVM is a kernel-based classification algorithm (BURGES, 1998), which has been shown to be effective in many types of applications. As reported, SVM is very efficient, stable (CASAGRANDE et al., 2017), and presents a good capacity of generalization (AKANDE et al., 2014).

The objective of SVM is to produce a model based on the training data. Thus, this model need to be capable of predicting the class of the sample using its attributes (HSU et al., 2003). The training set needs to be organized as sample-class $(x_i, y_i),\ i = 1, ..., l$, where $x_i \in R^n$ and $y \in \{1, -1\}^l$, and the SVM searches a solution for the optimization problem described by the Equation 5.1.

$$\begin{aligned} \min_{w,b,\xi} \frac{1}{2} w^T w + C \sum_{i=1}^{l} \xi_i \\ \text{Subject to } y_i(w^T \phi(x_i) + b) \geq 1 - \xi_i, \\ \xi_i \geq 0. \end{aligned} \tag{5.1}$$

SVM is based on the notion of the margin between different classes (KOTSIANTIS, 2007). The vector containing the training data x_i needs to be mapped to a higher dimensional space by the function ϕ. In

this new space, SVM tries to find a hyper-plane capable of separating the data with maximum margin aiming to reduce the generalization error by creating the widest possible distance between classes.

In their implementation, Samiappan et al. (2016a) used the library LIBSVM (CHANG; LIN, 2011). On their application, they proposed the use of radial basis function kernel. This Function was chosen because it can deal with non-linear data (HSU et al., 2003). The equation of the kernel is described by Equation 5.2.

$$K(x_i, x_j) = exp(-\gamma||x_i - x_j||^2), \gamma > 0 \tag{5.2}$$

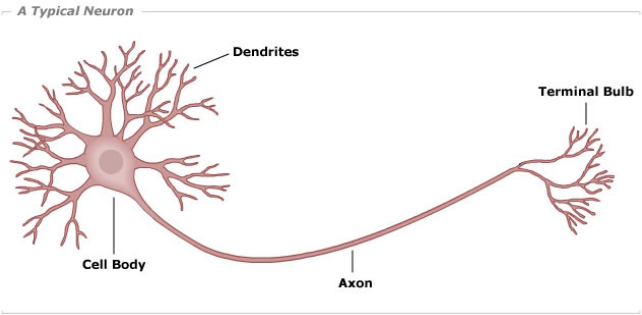
γ and C are the two major radial basis function parameters and they play an essential role in the final accuracy. The γ parameter affects the partitioning outcome in feature space. If this value is too high, it can results in overfitting. High values of γ causes overfitting, and low values causes underfitting. The parameter C represents the cost of penalty. High values of C distorts the accuracy rate, which will be high in the training phase and low in the testing phase. Low values of C may cause unsatisfactory accuracy (LIN et al., 2008). Aiming to optimize these parameters, Samiappan et al. (2016a) proposed the use of *Grid Search Algorithm*.

5.3 CLASSIFICATION BASED ON ANN

An ANN is a computational model based on the neural architecture of the brain. According to Haykin (2007), the interest in ANNs has been motivated by the growing knowledge about the human brain. As related by Rauber (1998), among all brain's capabilities, robustness and fault tolerance, ability to learn, processing of uncertain information, and parallelism are the characteristics that are especially attractive to be simulated in an ANN.

A human brain contains an enormous amount of nerve cells, also called neurons. Each of these neurons is connected to many other similar neurons, creating a very complex network of signal transmission. These neurons can be specialized in some task or not, but all of them are composed basically of a cell body, dendrites, and axons (Figure 18). Furthermore, neurons process signals in the same way (JR., 2004; AHARKAVA, 2010), i.e. the dendrites are responsible to bring information to the cell body and axons are responsible to take information away from the cell. The synapse is an electrochemical, not a physical, connection between one of the branches of the axon and the dendrite

Figure 18 – Biological Neuron.



Source: Image from The Pennsylvania State University (2017).

which has a small space between them (JR., 2004). Information from a neuron flows to another across a the synapse.

For a communication between neurons to occur, a signal must travel through the axon until the synaptic terminal, and after through the dendrites of the cell body. Signals are represented by electric impulses. These signals received from various neurons in the dendrites go to the cell body of the neuron that will sums them. If this sum is greater than some threshold, the cell fires and transmits a signal over its axon to other cells (AHARKAVA, 2010).

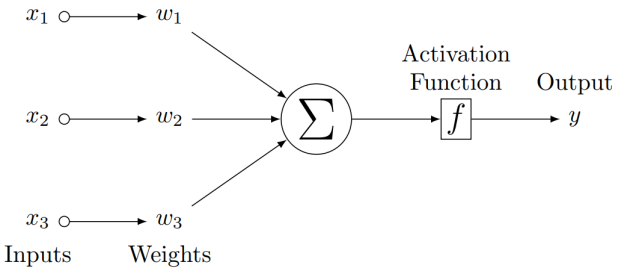
First ANN theory were presented in 1943 by Warren McCulloch and Walter Pitts (MCCULLOCH; PITTS, 1943), where the authors proposed the first artificial neuron model. These neurons could be further arranged into networks. However, since a learning rule was not defined at that time, these networks were not able to learn yet. Figure 19 shows a graphical representation of a simple artificial neuron. The neuron's output can be defined by the Equation 5.3.

$$y_k = f\left(\sum_{j=0}^{n} x_j w_{kj}\right) \tag{5.3}$$

Where y_k is the output from the kth neuron, n is the number of input, x_j is the jth input value, w_{kj} is the weight associated with the jth input and kth neuron, and f is the activation function.

Later, in 1949, the psychologist Donald Hebb designed the first learning rule for artificial neural networks (HEBB, 1949). In this case,

Figure 19 – Perceptron representation. In this image, y is the output signal, x_i is the value of the ith connection, w_i is the weight associated to the ith connection, and b is the bias associated.



Source: Image from authors.

the connection between two neurons should be increased if they were active simultaneously. After, the theory of ANN has evolved considerably with the perceptron's introduction in 1958 by Frank Rosenblatt (ROSENBLATT, 1958). He reported that adjustable connections could be used to train the ANN and classify certain types of patterns. However, it had a number of limitations, and as a consequence, the scientific community saw little interest.

With the advancement of technology and greater knowledge about the brain, new algorithms based on ANN have emerged over time. Mainly in the 80s, when several learning procedures for MLP were proposed, the interest in ANN has grown considerably. Nowadays, neural netwroks are being used in a huge number of problems by different scientific and industrial fields.

5.3.1 Multilayer Perceptron

Multilayer perceptron is one type of ANN. It was introduced with the advent of the back-propagation learning algorithm. Until that point, an architecture with more than one layer was not feasible since the learning algorithms were no efficient with that kind of architecture. A neural network of the type MLP consists of a system of interconnected neurons. Although it does not have a fixed architecture, such as in PNN and RBFN, these neurons follow a pattern of organization. In this case, they are arranged in three different types of layers:

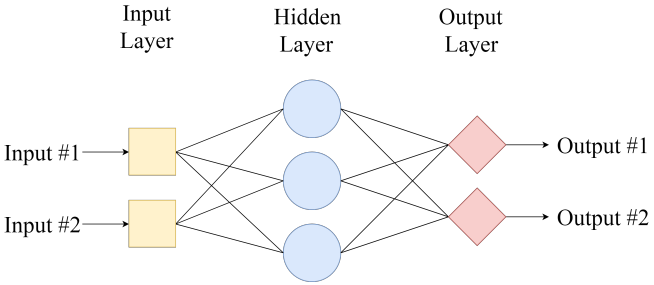
1. Input layer: The neurons in the input layer encodes the data from the external environment. Neurons in the input layer don't perform any type of computation. They only pass the input data to subsequent neurons;

2. Hidden layer: The neurons that compose the hidden layer receive the signal from the input or the previous hidden layer, process this data considering the Equation 5.3 and pass the output signal to the subsequent layer;

3. Output layer: The neurons from the output layer receive the signal from the hidden and compute the output.

In each MLP architecture, there is one input and output layer, and one or more hidden layers. The neurons and the layers are connected through synapses. That is, each connection has a weight that represents how strong is the link between the neurons. The MLP architecture can be fully connected, but this is not a requirement. An example of a MLP basic architecture is shown in the Figure 20.

Figure 20 – MLP fully connected architecture example. In this case, the architecture is composed of the input, one hidden, and output layer.



Source: Image from authors.

A Multilayer perceptron is a specific configuration of feedforward artificial neural network. That is, the output from a neuron always go forward until the output. For example, considering the Figure 20. First, as soon as the input data entry in the input layer, this data is passed to the first hidden layer. In the hidden layer, the input from the first layer is computed and passed to the subsequent layer. Finally, in the output layer, the data is processed again and output is presented.

In this technique, the process of learning through example is achieved by updating the synaptic weights associated with the connections between the layers. Currently, there are a large number of algorithms that aim to help to this process, where each one has its own specifications and can be more suitable for specific problems. In this work, the Levenberg-Marquard algorithm was used.

5.3.1.1 Levenberg-Marquard Learning Algorithm

First described by Kenneth Levenberg and Donald Marquardt in (LEVENBERG, 1944; MARQUARDT, 1963) and first reported as a possibility for ANNs in (HAGAN; MENHAJ, 1994), the Levenberg-Marquard (LM) algorithm provides a numerical solution to the problem of minimizing non-linear functions. This method can be seen as a blend of the Gauss-Newton and steepest descendent method, and fortunately, it inherits the speed advantage from the first and the stability from the second (YU; WILAMOWSKI, 2011). In other words, this method performs a combined training process, where around the area with complex curvature it switches to the steepest descent algorithm and in the areas where the quadratic approximation is feasible, it becomes the Gauss-Newton algorithm.

As demonstrated in recent comparison with other ANN learning algorithms (AYAT et al., 2012; YU; WILAMOWSKI, 2011; SEDHURAMAN; HIMAVATHI; MUTHURAMALINGAM, 2012), LM has high learning ability, which confirms the characteristics afromentioned. Ayat et al. (2012) state that LM has the best performance index and one of the shortest training time if compared to other five learning algorithms. Despite these facts, the authors highlighted that this algorithm presents high memory usage. Yu and Wilamowski (2011) and Sedhuraman, Himavathi and Muthuramalingam (2012) also state that LM algorithm is the most efficient among others. In addition to that, they reported that LM was the most stable, with more compact design, and less parameters.

As already described, the LM is very robust at dealing with respect to many problems. In addition to that, it has a simple formulation. Basically, it consists of solving the following equation:

$$J^t E = (J^t J + \lambda I)\delta \tag{5.4}$$

Where J is the Jacobian matrix for the architecture, λ is the damping factor, E is the error vector containing the simple error for each training case, $J^t J$ is the approximated Hessian matrix, and δ is the weight

updated vector.

The Jacobian is a N by M matrix, where N is the number of entries in the training set and M is the number of weights plus bias of the network. This matrix is calculated through all first-order partial derivatives of each output in respect to each weight. The overall architecture is described in the Equation 5.5.

$$J = \begin{bmatrix} \frac{\partial F(x_1,w)}{\partial w_1} & \cdots & \frac{\partial F(x_1,w)}{\partial w_m} \\ \vdots & \ddots & \vdots \\ \frac{\partial F(x_n,w)}{\partial w_1} & \cdots & \frac{\partial F(x_1,w)}{\partial w_m} \end{bmatrix} \tag{5.5}$$

Where $F(x_i, w)$ is the network function evaluated for the ith input vector of the training set using the weight vector w and w_j is the jth element of the weight vector.

The λ factor is adjusted at each iteration, where if the MSE (Equation 5.6) does not reduce, the weights are not updated and the λ factor is increased by a factor of 10. That is, if the reduction of MSE is fast, smaller values can be used, bringing the algorithm to the Gauss-Newton algorithm. Otherwise, the value increases and the algorithm gives a step closer to the gradient descent direction.

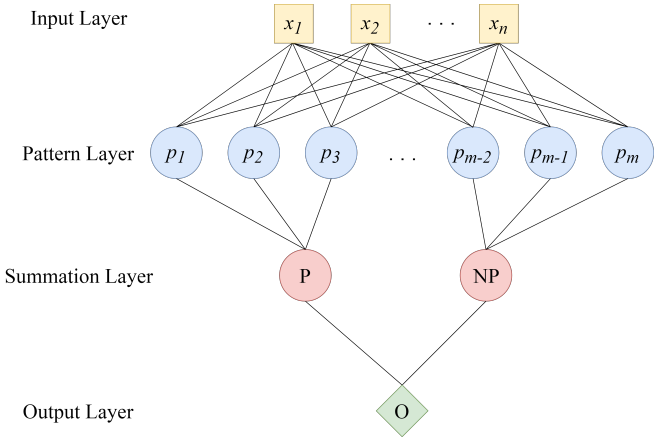
$$mse = \frac{1}{n}\sum_{i=1}^{n}(\widehat{Y_i} - Y_i)^2 \tag{5.6}$$

Where $\widehat{Y_i}$ is the vector denoting the output value for each training case, Y_i the vector representing the true values, and n the number of samples in the training set.

5.3.2 Probabilistic Neural Network

Specht (1990) presented the PNN classifier, that is a specific configuration of feedforward artificial neural network ANN. According to Specht, there are some advantages of using PNN over other classification algorithms, among them, it has a well-defined architecture and, hence, straightforward implementation; it has an instantaneous training process, where the training cases are incorporated into the network as new nodes; and it is robust with respect to noise in the input. Moreover, under certain easily met conditions, it can approach a Bayes optimal result. It is appropriate to say that the PNN is in a special class of ANN for classification problems.

Figure 21 – Illustration of our PNN's architecture for classification. The input layer has n nodes, one for each input parameter. The pattern layer is fully connected to the input layer, and its nodes represent the training cases. The summation layer averages the results from pattern layer for each of the two categories (positive P or negative NP). And the output layer decides to which category (i.e., with or without *Phragmites*) the input corresponds.



Source: Image from authors.

A typical PNN has four layers: input, pattern, summation, and output (Figure 21). The input layer maps the n input parameters from the input data to the PNN.

The pattern layer has one node p_i for each training case and it is fully connected with the input layer. Computation in each pattern node is given due to the following equation:

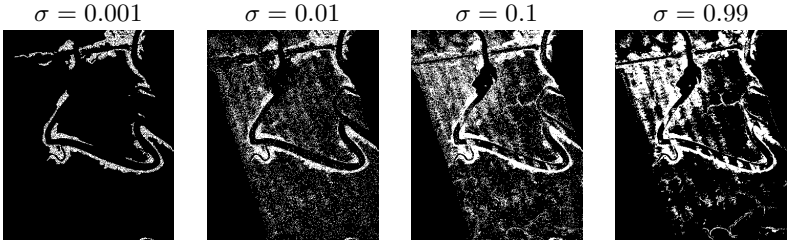
$$p_i = \frac{1}{(2\pi)^{\frac{n}{2}}\sigma^n} exp\left[-\frac{(v-v_i)^T(v-v_i)}{2\sigma^n}\right], \tag{5.7}$$

where v_i is the vector representing the corresponding training case and σ is a value of the function that determines the spread of the function.

As demonstrated in Figure 22, this value plays an essential role in the classification algorithm. As presented, a higher value, that is not very selective, will benefit the predominant class. The higher value, that is very selective, results in misclassification of the objects with the closest texture to the target class. In the other side, with a lower

value, all blocks are assigned with the predominant object class. Consequently, the optimal sigma value selection is essential for the process.

Figure 22 – Output map from PNN using data output from WTA with four different σ for the set II.



Source: Images from authors.

Further, the pattern nodes can be divided into subgroups according to the training cases' category. Two categories were defined in this work for the PNN: *Phragmites* (P) for blocks with *Phragmites* and not *Phragmites* (N) for blocks without *Phragmites*.

The summation layer computes the maximum likelihood of input v being classified into each one of the categories. For such, each node in the summation layer corresponds to one distinct category and it is connected to the corresponding subgroup of nodes in the pattern layer. In fact, the summation nodes perform an averaging of the results in the connected pattern nodes as follows:

$$s_c = \frac{1}{N_c} \sum_{i}^{N_c} p_i \tag{5.8}$$

where p_i in the sum represents the results from pattern nodes of category c and N_c is the total number of pattern nodes in this category. If the probabilities for each category are the same, the decision layer unit classifies the pattern v following the Bayes' decision rule based on the output of all summation layer neurons as follows:

$$\hat{C}(v) = argmax \{s_c\}, \ i = 1, 2, \cdots, q \tag{5.9}$$

Where $\hat{C}(v)$ denotes the estimated class of the pattern w and q is the total number of classes in the training samples.

5.3.3 Radial Basis Function Network

Radial basis function network was proposed as a new paradigm of simpler ANN architectures with only one hidden layer that could reduce the necessary time to train a architecture considering the traditional back-propagation learning algorithm. This method was first described by (POWELL, 1987) to solve the interpolation problem in a multi-dimensional space requiring as many centers as data points. Later, Broomhead an Lowe in (BROOMHEAD; LOWE, 1988) removed this restriction, allowing the use of a smaller number of centers and increasing the number of possible applications.

Although it is possible to define RBFN as a special case of MLP, as described by Xie, Yu and Wilamowski (2011), there are four main differences between them:

1. RBFN has a well defined architecture;
2. RBFN performs as local approximation networks. MLP performs global approximation;
3. it is essential to set correct initial states for RBF networks while MLP networks use parameters generated randomly;
4. RBFN clusters are separated by hyper spheres; while MLP uses arbitrarily shaped hyper surfaces for separation.

Although RBFN and PNN are similar as well, there are some differences between both methods:

1. RBFN has only one hidden layer and PNN has two;
2. Although it is possible to consider a neuron per test data, the use of clustering algorithms to reduce the number of neurons is a common approach;
3. RBFN may have a training phase;

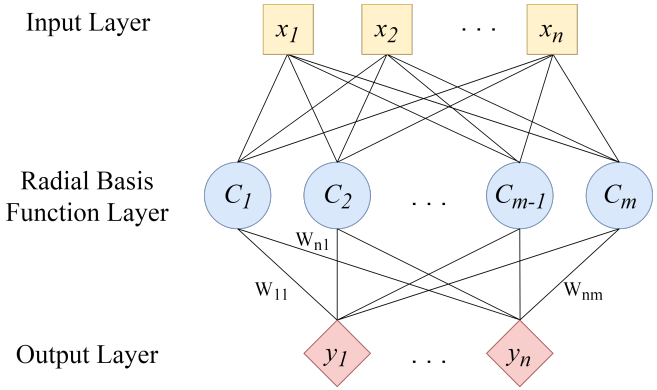
As a consequence of the described differences, this technique has some advantages and disadvantages over other techniques. As concluded by (XIE; YU; WILAMOWSKI, 2011), this technique is more robust and tolerant to noise input data set if compared to ANN. However, according to Zhao, Huang and Guo (2003), PNN presents better noise tolerance than RBFN mainly because there exists more than one hidden center vector for each pattern class. Zao et al., Xie et al., and Mirzaei et

al. (ZHAO; HUANG; GUO, 2003; XIE; YU; WILAMOWSKI, 2011; MIRZAEI et al., 2011) showed that RBFN is the worst choice for classification systems. However, as reported in (MEDAPATI et al., 2011) this may not be true for all problems. In addition, the authors from (XIE; YU; WILAMOWSKI, 2011) described that for function approximation problems, RBF networks are specially recommended for surface with regular peaks and valleys, since efficient and accurate design can be obtained

5.3.3.1 Radial Basis Function Network architecture

Commonly the RBFN method has three layers: Input, radial basis functions, and output. The n input parameters from the input data are mapped by the input layer. Figure 23 shows the RBFN's overall architecture.

Figure 23 – Illustration of our RBFN's overall architecture. The input layer has n nodes, one for each input parameter. The hidden layer, here identified as radial basis function layer, is fully connected to the input layer, and its nodes represent the identified centroids. This layer is composed of radial basis function neurons that is a specific type of artificial neuron. The output sums the weighted outputs from the hidden layer for each category (in this case, just one category with possible classes: *Phragmites* or not *Phragmites*).



Source: Image from authors.

In the radial basis function layer, each node represents one center. This layer is fully connected to the input layer. There are different

possible choices of activation functions for this layer, but in this work it was used the most popular, that is based on the Gaussian. The Gaussian activation function is described below:

$$rbf_i = \frac{1}{\sigma\sqrt{2\pi}} e^{-\frac{\|x_i - c_i\|^2}{2\sigma}} \tag{5.10}$$

Where x_i is the vector representing the corresponding training case, c_i is a vector representing one center, and σ is a value of the function that determines the spread of the function. In the RBFN algorithm, the σ also plays an essential role in the final classification accuracy.

The output layer basically sums the weighted outputs from the radial basis function layer. The final equation for this layer is described below:

$$y_k = \sum_{i=0}^{n} rbf_i w_{ki} + b_i \tag{5.11}$$

Where w_{ki} denotes to the weight of each connection and b_i the bias for each output neuron.

5.3.3.2 RBFN Learning

As demonstrated in (SCHWENKER; KESTLER; PALM, 2001), there are many possible approaches to train RBFNs. In this work, a two-stage training algorithm based on k-means clustering and in the normal equations method was used.

K-means clustering algorithm is an unsupervised classifier that is commonly used in the data mining and image recognition area. The main objective of this algorithm is to define the k centers based on clusters. In the RBFN, each of these clusters is represented in the radial basis function layer by an artificial neuron.

This is an iterative algorithm that can be divided basically into four steps: Before the loop starts, the groups are randomly labeled. After, the distance between each data and the k clusters is calculated. Next, the data is labeled according to the nearest centroid. The last step is to calculate the average of each cluster separately. Based on these values, the new positions of the centroids are found. This process repeats until the stop criteria are satisfied. In this case, the stop criteria used are the maximum number of iterations, defined as 100, or

a maximum number of data moving between clusters in one iteration, defined as 1% from the total number of vectors. There are some ways of calculating the distance between each data and the clusters, such as squared euclidean distance, Hamming distance, Mahalanobis distance, and among others. In this work the Euclidean distance was chosen (Equation 5.12).

$$d(x, c) = (x - c)(x - c)' \tag{5.12}$$

Where x is a vector from the input data and c is the centroid, that represents the average value from all points in the cluster.

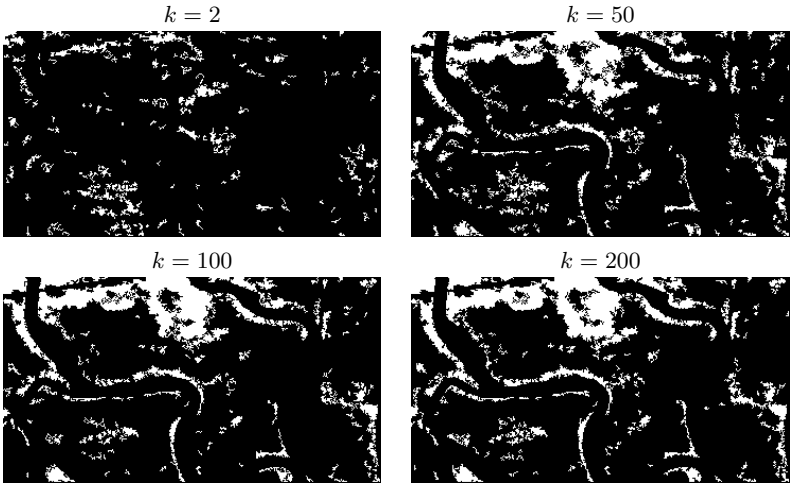
In addition to the σ value, the number of k is also a parameter that can change drastically the final accuracy and computational cost necessary to generate the final map. Consequently, the optimal sigma value selection is essential for the process. Figure 24 shows examples of maps generated with a different number of centers are shown. From this figure, it is possible to conclude the essential role that the number of centers has for the map generation. In the first situation, with $k = 2$, the mapping has practically no representation in reality. Few *Phragmites* sites were identified and many are outside the correct area. However, in the last situation with $k = 200$, it is possible to perceive that the true positive rate has increased considerably. From this figure, we can also verify that there is some evolution with the number of k, but this is not true for all situations. For example, a $k = 101$ will not necessarily present a map with higher accuracy than a $k = 100$.

After this process, the weights are updated using the normal equations. This process can be simplified into three steps: First, for every training case in the data set, the activation value of the RBF neurons is calculated. After, for each category in the problem, a binary vector identifying the position of all training cases correspondent to the category is generated and the optimal weight is identified using normal equations (Equation 5.13).

$$\beta_c = (X^T X)^{-1} X^T y = X^+ y \tag{5.13}$$

Where β_c is the new vector for each category, X is the activation matrix, y is the binary vector for each category, and X^+ is the Moore-Penrose Matrix Inverse.

Figure 24 – Output map from RBFN using data output from WTA with four different k for the set I.



Source: Images from authors.

5.4 HYPERPARAMETER TUNING

Hyperparameter optimization is an essential step before the classification. Many of the machine learning and data mining algorithms are sensitive to the setting of parameters, which is considered a task of a human expert (MATUSZYK et al., 2016). The most used method to define these parameters is the trial and error, where the main objective is to test the greatest amount of possibilities until finding a classifier with good generalization capacity. However, it is known that this method is time-consuming and depends directly on the knowledge and prior experience of the expert for designing the ANN. In addition, setting a non-optimal parameter can drastically reduce the accuracy or cause a non-convergence situation.

Another aspect associated with the importance of this step is the necessity to marginalize the influence of the human choice of the parameters. When this process is done manually by a human, the comparison is not feasible because it is not known whether the error difference is actually being caused by the algorithm or by the parameter. The use of an objective and fair hyperparameter tuning algorithm is a

way to reduce the influence of this highlighted question.

In this work, two algorithms were considered for this task. In the PNN tuning process, it was used the grid-search algorithm. In the MLP and RBFN, it was used GA. These choices were based on the size of each algorithm search space, defined by the number and size of each set of combinations. Considering that in the first classifier the search space is considerably smaller than in the second and third algorithm, it is feasible the use of an exhaustive searching such as grid-search. This fact is not true for the second and third case, that has a bigger search space as will be explained in the next sections.

However, prior to any of these algorithms, the 10% selected for this step must be redivided into training and test set. In this case, 80% from these cases are randomly selected for the first ground and the other part for the second one. In this case, understanding that the distribution of the training set can influence in different ways the final result of the solution, the number of blocks representing *Phragmites* ranged from 30 to 75% in the first set and from 20 to 45% in the second set. The rest of the training set is completed with blocks representing other objects. The test set consists of all the blocks that are not in the train set.

5.4.1 Grid-Search Algorithm

As described in the previous section, this exhaustive search algorithm was chosen to perform hyperparameter tuning in the PNN algorithm. This is a consequence of the search space size ($\overrightarrow{\theta}$) of PNN if compared to the $\overrightarrow{\theta}$ defined to the MLP method.

Hsu, Chang and Lin (2003) stated that the grid-search is straightforward but seems naive. In fact, there are several advanced methods which can save computational cost by, for example, approximating the MSE rate. However, there are three reasons for choosing this algorithm in this situation: First, this is a way of ensuring that all desired situations are tested. Second, its performance is similar to complex algorithms in narrow search spaces. Third, the grid-search can be easily parallelized because the parameters are independent.

This algorithm consists of exhaustive searching through a manually defined subset of the hyperparameter space. The two most common types of this algorithm are the "Cartesian" and the random grid search. In the first type, the algorithm test each possible combination of hyperparameter values. In the second, the algorithm will sample uniformly

from the set of all possible hyperparameter value combinations in the search space. In this work, the Cartersian type was used. The fitness of each combination was defined using MSE, defined in the Equation 5.5.

After this process of testing all combinations, the best combination is identified and used for map generation.

5.4.2 Genetic Algorithm

GA, that was first proposed by John Holland in the mid-1970s, was based on the evolutionary biology theory presented by Charles Darwin. This method was developed with the goal of becoming a powerful results optimization algorithm. The main advantage presented by GA, when compared to other optimization methods, is that GA searches for the best results based on individuals' potential of the population (GREFENSTETTE, 1986).

5.4.2.1 Genetic Algorithm and ANN-MLP

When proposing the use of an ANN MLP to solve a possible problem, the only certainty is that there should be an input and an output layer, and that the number of neurons in each of these layers is specified by the number of input and output parameters used to model the system. Therefore, it is not possible to theoretically determine the exact number of hidden layers and neurons in each hidden layer for each problem.

The activation functions generally are chosen based on the type of data that is used to train the ANN. For example, the linear function is almost always used in the input layer and some sigmoid function is used in the hidden layer. However, these guidelines do not guarantee the ideal choice for the problem in analysis.

The training algorithm influences directly in the training speed, ANN performance and the computational cost to train. Therefore, the choice of the algorithm does not influence the generalization capacity.

In summary, there is no specific methodology to define the number of hidden layers, number per layer, activation function and the training function. However, it is clear that essentially the number of neurons and hidden layers, and activation functions are the main characteristics that can drastically modify the generalization level of the

system.

According to Akande et al. (2014), considering that the objective of ANN MLP optimization is multimodal in nature, this technique tends to converge to a local minimum. In addition, the choice of architecture and training parameters for ANN MLP is a task that directly impacts the performance of the classification system and the generalization capacity. As related by Haykin (2011), a architecture wrongly defined also tends to converge to a local minimum.

Understanding the critical nature of the problem and the amount of data obtained for analysis in this work, the use of trial and error becomes an undesirable option and can reduce drastically the accuracy of the proposed system. In addition, the use of an exhaustive search algorithm such as grid search can result in a high computational cost due to the search space generated by all ANN MLP variables. Genetic algorithm can be an alternative to search the "optimal" solution in the search space mainly by its evolutionary capacity.

Recently many papers proposed the use of GA with ANN to perform search and optimization tasks, as demonstrated in (BENARDOS; VOSNIAKOS, 2007; PATER, 2016; DYKIN et al., 2015). Such combination of computational techniques has already demonstrated a good performance finding the optimal result in the proposed research space.

There are different ways of GA interacting with an ANN, as described in (SCOTT; JAMES; ALI, 2006), such as:

- ANN contributing with GA: The neural network creates the first population while GA optimizes;
- GA contributing with an ANN: In this case, GA can contribute by selecting parameters, learning rate, training function, activation function, weights, bias, among others.

In this work, the second type of interaction was used.

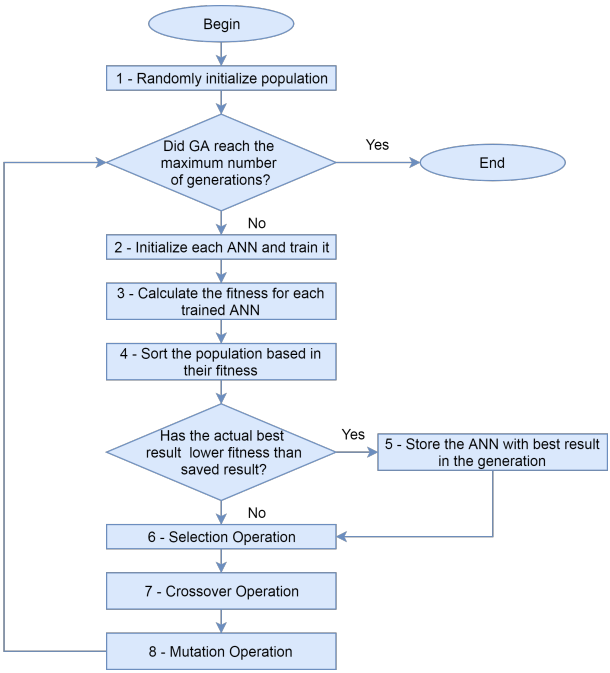
5.4.2.2 Optimization Algorithm

The optimization system can be described by the execution flow represented in Figure 25. In general, this system is divided into specific functions that are responsible for different operations in the optimization process. These include: generate the individuals, initialize each ANN and train it, quantify the trained classifiers, sort the generation based on the fitness, eliminate individuals considered weak, create indi-

viduals through crossover operation, and mutate the generation under analysis.

First, a random initial population is generated using a dynamic chromosome sequence. Each individual that composes the population was organized following the structure that is shown in Figure 26.

Figure 25 – Genetic Algorithm-Artificial Neural Network algorithm flowchart composed by eight process and two decision. Three from these eight process are specific from GA algorithm: selection, crossover, and mutation.

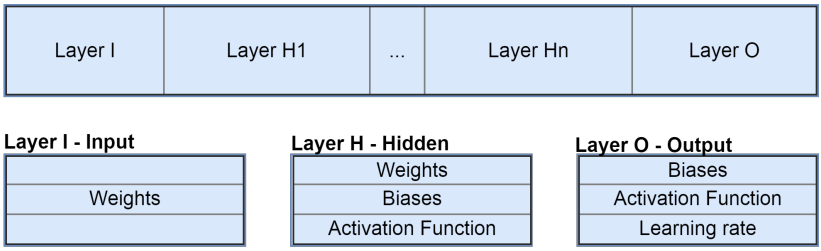


Source: Image from authors.

The chromosome that represents the input layer stores all weights between the input neurons and the neurons that compose the first hidden layer. The chromosomes that symbolize the hidden layer(s) are organized between the genes that represent the bias for the actual layer, the weights from the actual layer to the next layer, and the activation function for the actual layer. The output layer genes are divided

between the bias and activation function for the actual layer, and the learning rate for the ANN that represents the architecture.

Figure 26 – Individual general structure. The input layer is represented by the chromosome I. The hidden layers are represented by the chromosome H and can range from 1 to n where n represents the number of hidden layers. The chromosome H represents the output layer. Each chromosome type represents different features from the architecture.



Source: Image from authors.

Both, the number of internal layers and the number of neurons per layer, were generated randomly. For each layer, an arbitrary activation function was defined. Each layer has its function defined within a group of three distinct functions: linear, Log-sigmoid transfer function (as described in 5.14) and Hyperbolic tangent transfer function (described in 5.15).

$$logsig(n) = \frac{1}{1 + e^{-n}} \tag{5.14}$$

$$tansig(n) = \frac{2}{1 + e^{-2*n}} - 1 \tag{5.15}$$

The biases of all layers and the weights of the layers that have logistic sigmoid and tangent sigmoid activation function were defined using the Nguyen-Window method. The weights of the layers that have a linear activation function were randomly defined between -1 and 1. The learning rate was also randomly defined between 0 and 1. Then, the previously defined data is structured in an object representing an ANN for later training using LM algorithm.

After the training process, the test set, which is composed of data not known by the classifier, is used to assess the generalization performance of the model generated. The MSE obtained in the validation

process is then stored for later ranking of the individuals.

In the fourth step, the values obtained after the training and validation process are used to evaluate the entire population. After, the best-fit individual of the generation is saved as the best individual if its fitness is lower than the actual best-fitness individual.

The evolutionary process happens between the steps 6 and 8. In the sixth step, 50% of the individuals are selected based on the ranking made by the fitness obtained. After the selection operation, the individuals that were not eliminated generate new individuals through the crossover operation. In this operation, a pair of individuals is selected sequentially, where the first half of the first individual chromosome and the second half from the second individual chromosome are used to create a new individual for the next generation. However, considering the fact that there was a change in the number of neurons between the first half and the second half of the new individual, new values for initial weights and bias are generated following the procedure and search space described in the individuals' generation. In the eighth step, 10% of individuals are selected for the mutation operation. Being selected, the individual has equal changes to suffer some kind of mutation in the learning rate, activation functions, number of neurons in a specific layer or number of layers. Such operation ensures that if any chromosome is lost through selection and crossover, it can reappear in future generations.

After this process, it is verified if the algorithm reached the number of generations previously defined. If yes, the algorithm presents the best individual found. If not, the process returns to the step 2.

5.4.3 GPU Implementation

As well as in the feature extraction process, a wide range of opportunities for parallelization is created by the CPU implementation proposed in this work. Two steps must be performed in this process: Hyperparameter Tuning and map generation. For each of these steps, different approaches can be conducted aiming to optimize the GPU implementation.

In the first case, the search step-by-step algorithm could be optimized, as demonstrate in (JAROS; POSPICHAL, 2012; CARPENTER, 2009). Jaros and Pospichal (2012) proposed a fair GA algorithm comparison between the CPU and GPU implementations. Carpenter (2009) reported a SVM and grid-search GPU implementation. Another possi-

bility would be a classifier GPU implementation, as reported in (PHAUDPHUT; SO-IN; PHUSOMSAI, 2016; BRANDSTETTER; ARTUSI, 2008; RAIZER et al., 2010), where Phaudphut, So-In and Phusomsai (2016) proposed a PNN GPU implementation, Brandstetter and Artusi (2008) a RBFN implementation, and Raizer et al. (2010) a ANN-MLP implementation. As described in the feature extraction GPU implementation, another feasible option would be the use of two levels of parallelization. In this option, both the optimization process and the training and prediction process could be processed in parallel.

The map generation algorithm offers a wide range of opportunities too. First, as already described, the specific classifier execution could be modified aiming a GPU implementation. Since this problem has a large number of instances, a parallel execution would be a option too. As in the other cases, the combination of these possibilities creates new possibility of implementation. Aiming to reduce the amount of modifications of the algorithms, in this work the GPU parallelization was performed on the search algorithm in the first step and in the instances in the second step.

In the hyperparameter tuning step, each image block is processed as follows:

- Training and validation set are read into host memory;
- Data is copied to the global memory device;
- Grid is defined for the hyperparameter tuning step. Each grid block represents one possible combination.
- The chosen parameters are copied back to the host.

For the grid-search algorithm, only one kernel is used. In this case, each combination is one block from the grid. The training and fitness calculation process for each combination is linear.

In the GA algorithm, the algorithm's specific operations were also implemented in the GPU. Each individual from the population is represented by a grid block, that is, the individuals are parallelized in this process. As in the grid-search algorithm, the training and fitness calculation process is linear.

After, in the hyperparameter tuning step, each image block is processed as follows:

- Features and parameters are read into host memory;
- Data is copied to the global memory device;

- Grid is defined for map generation step. Each grid block represents one instance.
- The output classes are saved in the host memory.

6 RESULTS AND DISCUSSIONS

6.1 IMPLEMENTATION SPECIFICATIONS

The main parameters defined for this work are shown in this section.

6.1.1 Hyperparameter Tuning Search Space

The search space was defined based on the main characteristics of each classifier algorithm. Considering that just the σ was optimized, PNN had the smallest search space. Its definition can be found in the Table 4.

Table 4 – PNN's search space totaling 90 combinations.

Parameter	Step	Range
σ	0.01	0.05-0.95

Source: From authors.

As described before, GA algorithm had to be employed to find optimal parameters for RBFN and MLP algorithms. Search space size of each algorithm created this necessity to use another search algorithm. Two parameters were optimized for the RBFN algorithm: the number of centers (k) and the σ. The search space used in the experiments can be found in the Table 5.

Table 5 – RBFN's search space.

Parameter	Range
σ	0.05-0.95
k	2-802

Source: From authors.

Mainly because the ANN-MLP algorithm does not have a well defined architecture, this algorithm had the largest search space when compared to the other methods. One problem associated with this

issue was the definition of the search space. The bias and activation function parameter search space is known and well defined. However, the ideal number of hidden layers and neurons per hidden layer is not known. Considering this context, an experiment was conducted aiming to identify the convergence capacity with different MLP architectures.

In this experiment, the grid search algorithm was used. In each combination of hidden layers and number of neurons per hidden layer, a GA optimization process was employed. In this case, 10% from the Set I blocks were used totalling 6654 cases. From this number, 10% was experimentally selected as training case. SFTA with $nt = 6$ was used to extract features from the blocks. In the optimization, 100 individuals were tested by 100 generations. The bias value ranged from 0 to 1 and the activation function was chosen between linear, hyperbolic tangent sigmoid, and Log-sigmoid. The grid search space used in this experiment is defined in the following table:

Table 6 – Search Space for MLP experiment totalling 100 combinations.

Parameter	Step	Range
Hidden Layers	2	1-19
Max Neurons per Layer	2	1-19

Source: From authors.

The best MSE from each combination was found. These values are shown in the Table 7.

Table 7 – Results from MLP experiment.

	Maximum Number of Neurons per Layer										
Hidden Layers		1	3	5	7	9	11	13	15	17	19
	1	0,085	0,086	0,086	0,085	0,083	0,081	0,081	0,084	0,079	0,086
	3	0,084	0,081	0,077	0,081	0,088	0,092	0,082	0,085	0,080	0,083
	5	0,088	0,084	0,086	0,092	0,089	0,121	0,083	0,080	0,085	0,091
	7	0,085	0,500	0,121	0,500	0,093	0,094	0,087	0,092	0,079	0,095
	9	0,086	0,498	0,500	0,500	0,500	0,500	0,084	0,139	0,113	0,500
	11	0,098	0,500	0,499	0,499	0,500	0,500	0,481	0,500	0,500	0,346
	13	0,086	0,500	0,500	0,500	0,500	0,500	0,087	0,236	0,500	0,500
	15	0,500	0,500	0,500	0,500	0,500	0,500	0,500	0,208	0,500	0,500
	17	0,236	0,500	0,500	0,500	0,500	0,500	0,500	0,500	0,500	0,500
	19	0,500	0,500	0,500	0,500	0,500	0,500	0,449	0,500	0,500	0,500

Source: From authors.

From the values presented in the Table 7, it is possible to conclude that the convergence capacity reduced considerably when the number of hidden layers was above 9. In addition, within the search space defined for the number of neurons per layer, it is noticed that this parameter did not result in abrupt changes in the MSE value. With this information, the GA total search space was defined as follows:

Table 8 – Search space for ANN-MLP.

Parameter	Range
Bias	0-1
Activation Function	Linear, Log-Sigmoid, Hyperbolic Tanget Sigmoid
Max Hidden Layers	1-10
Max Neurons per Layer	1-19

Source: From authors.

6.1.2 System Assessment

This work general objective is to increase the overall accuracy and reduce the time required for *Phragmites* mapping process when compared to the state of the art. Considering this fact, specific methodologies were proposed for the accuracy and computing time assessment.

6.1.2.1 Accuracy Assessment Methodology

Aiming to evaluate the system's performance, the GRM was used to calculate the confusion matrix. This is a specific table commonly used in the machine learning area that is often used to describe the performance of a classification model. Mainly in this case where the classifiers can be defined as binary classifier, the confusion matrix is one of the simpler ways to present the system's performance. An example of confusion matrix is shown in Table 9.

This matrix can be further used for the calculation of other statistical parameters. These parameters are intended to describe specific situations presented in the final result. The parameters used to evaluate the performance of each combination are the kappa coefficient (κ),

overall accuracy (OA), agreement (A), commission error (CE), and omission error (OE) (JANSSEN; VANDERWEL, 1994).

Table 9 – Confusion Matrix for Binary Classifiers, where P represents *Phragmites*, NP not *Phragmites*, TP is the number of true positive cases, FP is the number of false positive, FN is the number of false negative, and TN is the number of true negative.

		True Condition	
		P	NP
Predicted Condition	P	TP	FP
	NP	FN	TN

Source: Adapted from Janssen and Vanderwel (1994).

The OA is the rate of correct pixels among all pixels and can be calculated by the following equation:

$$OA = \frac{TP + TN}{TP + TN + FP + FN} \tag{6.1}$$

The A is the rate of correct pixels within a given category, in other words, it represents the probability of a reference pixel being correctly classified. It can be calculated by the Equation 6.2.

$$A_c = \frac{P_c}{P_c + N_c} \tag{6.2}$$

Where c is the indication of class. For example, in the A_{np} case, the P_c value represents the false positive and N_c the false negative.

The CE represent pixels that belong to another class but are labeled as belonging to the class and can be calculated using the following Equation:

$$CE_c = \frac{T_c}{T_c + F_c} \tag{6.3}$$

Where true and false are the values found for the predicted condition. For example, in the CE_P case, the T_c value represents the true positive, and the F_c represents the false positive.

The OE (Equation 6.4) represent pixels that belong to the truth class, but fail to be classified.

$$OE = 1 - A_c \tag{6.4}$$

According to Viera and Garrett (2005), the κ is the most com-

monly used statistic in remote sensing for studies that measure agreement between two or more learners. This coefficient can be calculated with the following equation:

$$\kappa = \frac{OA - p_e}{1 - p_e} \tag{6.5}$$

Where:

$$p_e = p_{Yes} + p_{No} \tag{6.6}$$

$$p_{No} = \frac{TN + FN}{TP + TN + FP + FN} * \frac{TN + FP}{TP + TN + FP + FN} \tag{6.7}$$

$$p_{Yes} = \frac{TP + FP}{TP + TN + FP + FN} * \frac{TP + FN}{TP + TN + FP + FN} \tag{6.8}$$

6.1.3 Computational Time Assessment Methodology

The analysis of the necessary time to generate a map was divided into two steps: (i) Feature extraction and (ii) hyperparameter tuning and classification. In (i), the necessary time to run each algorithm was computed 100 times. In the second case, considering that the optimization and classification process are much more time consuming, the time was computed 5 times per algorithm. From these values, mean value was computed.

6.1.4 Experimental Setup

The maps were first generated in a MATLAB implementation. These experiments were performed on a 2.5GHz Intel i5-2450M on 64-bit Microsoft Windows operating system with 8 GB of RAM.

A C implementation was conducted targeting a fair comparison between the necessary time to generate a *Phragmites* map in CPU and GPU. The CPU experimental is presented in the following table:

Table 10 – CPU Experimental Setup.

CPU Version	Intel i5-4670K
CPU Core	3.4GHz
Memory Size	16GB
Operating System	Ubuntu 16.04

Source: From authors.

This C implementation was further used as base for the CUDA implementation. The GPU experimental setup can be found in the following Table:

Table 11 – GPU Experimental Setup.

GPU Version	GeForce GTX 770
#Cuda Cores	1536
Core Clock	1046MHz
Memory Type	256-bit GDDR5
Memery Bandwidth	224Gbps
Memory Size	2GB

Source: From authors.

GPU experiments were conducted on the same host described for the CPU.

6.2 ACCURACY ASSESSMENT

In our experiments Site I and Site II were considered. A map was generated for each combination of Site-Solution. In these maps, *Phragmites* agreement is outlined in red, omission cases outlined in green, and commission cases in blue. In addition, Class-specific classification accuracies (in percentage), agreement (in percentage), OA (in percentage), and kappa statistic were calculated for each combination. After, these values were used in classifier comparisons.

6.2.1 Grey Level Co-Occurrence Matrix

Table 12 shows the statistical results for the three classifiers and the state-of-art result. Note that the κ values from GLCM-PNN and GLCM-MLP are above 0.8. According to Viera and Garrett (2005), this value represents almost perfect agreement. In the GLCM-RBFN, the κ is between 0.61 and 0.8, which means substantial agreement.

Table 12 – Site I - GLCM Texture Features.

	PNN		MLP		RBFN		SVM	
Class	P	NP	P	NP	P	NP	P	NP
OE (%)	19.6	2.3	16.6	2.6	10.4	4.7	19.0	3.0
CE (%)	14.3	3.3	15.6	2.8	23.8	1.8	18.0	3.0
A (%)	80.4	97.7	83.4	97.4	89.6	95.3	80.0	96.0
κ	0.8018		0.8118		0.7910		0.7700	
OA (%)	95.2		95.3		94.5		94.0	

Source: From authors.

Table 13 shows the statistical results for site II. In this case, only the κ value from GLCM-PNN is above 0.8. In the other solutions, the κ is between 0.61 and 0.8.

Table 13 – Site II - GLCM Texture Features.

	PNN		MLP		RBFN		SVM	
Class	P	NP	P	NP	P	NP	P	NP
OA (%)	10.8	0.9	19.5	2.0	31.4	0.9	28.4	2.5
CE (%)	11.7	0.84	24.3	1.5	14.9	2.4	31.0	2.2
A (%)	89.2	99.1	80.5	98.0	68.6	99.1	71.6	97.5
κ	0.8790		0.7626		0.7426		0.6700	
OA (%)	98.4		96.7		96.86		95.6	

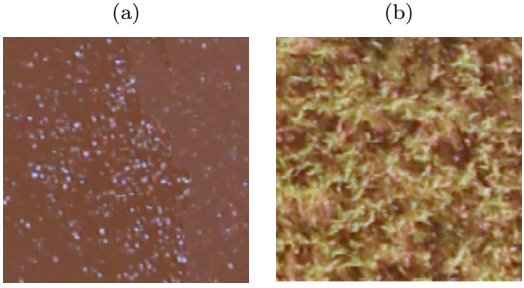
Source: From authors.

Although the GLCM-PNN had a better performance in few parameters, the GLCM-MLP solution generated the most reliable map for Site I. This difference is practically negligible if we consider the large number of variables involved in the proposed system. The good aspects associated with the GLCM-RBFN results were the best *Phrag-*

mites agreement and lowest commission error. Although the GLCM-RBFN kappa value is in the same range presented by the state-of-art, its results are slightly better than the presented by the state-of-art. The same is not true in a comparison between the GLCM-MLP and the GLCM-RBFN result. GLCM-MLP combination performed better in all statistical parameters and presented a considerable improvement in the reliability level of the generated map.

This relation between the techniques changes in the Site II. In this case, the best result was presented by GLCM-PNN and there is a considerable difference between GLCM-PNN and GLCM-MLP statistical parameters. Furthermore, there are some differences between GLCM-RBFN results for Site I and II. Although GLCM-RBFN not *Phragmites* agreement is one of the highest, the *Phragmites* agreement is the lowest. Another fact to be highlighted is the not *Phragmites* agreement presented by the GLCM-RBFN combination. Althought this is not the best result, it presented the highest not *Phragmites* agreement. In addition, despite a lower κ if compared to the GLCM-MLP solution, the overall accuracy is higher. Both facts can be explained by the missclassification of considerable stands.

Figure 27 – Comparison between a river sunlight reflection (a) and *Phragmites* (b) blocks.



Source: Images from authors.

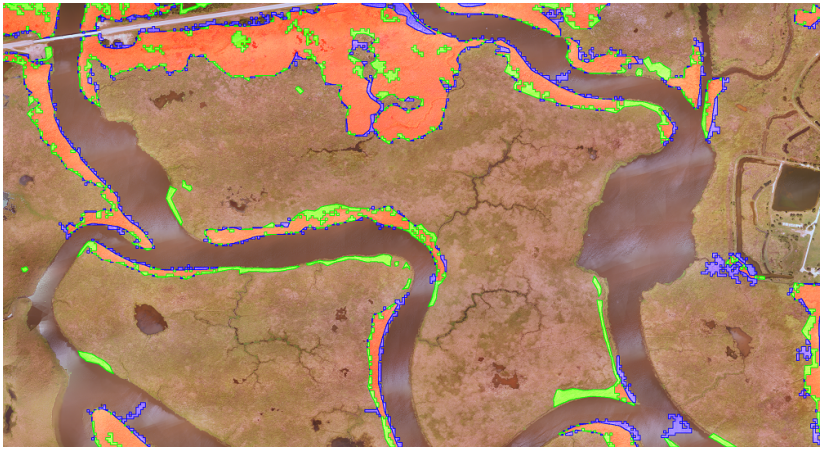
As shown in Figures 28, 29, and 30 from Site I and Figure 31 from Site II, the main area of misclassification is with respect to blocks that represent locations of transition between P and NP. This is a block processing drawback because in some of these blocks the non-*Phragmites* object comprises higher area than *Phragmites*, and, as consequence, the distance found by the classifier between the block's pa-

rameters and NP was smaller than with P. In addition, in these cases regardless of the class defined by the classifier, part of the block will be a source of missclassification. River sunlight reflection is a common source of error for the three approaches too. This can be explained mainly by the granularity presented in both cases, as shown in the Figure 27.

In addition, intermixing of vegetation was a smaller source of error for the three methods. This was already expected, because in several related works the problem related to the vegetation mix was discussed. In both Sites, the intermixing of threes and *Phragmites* was the main problem found.

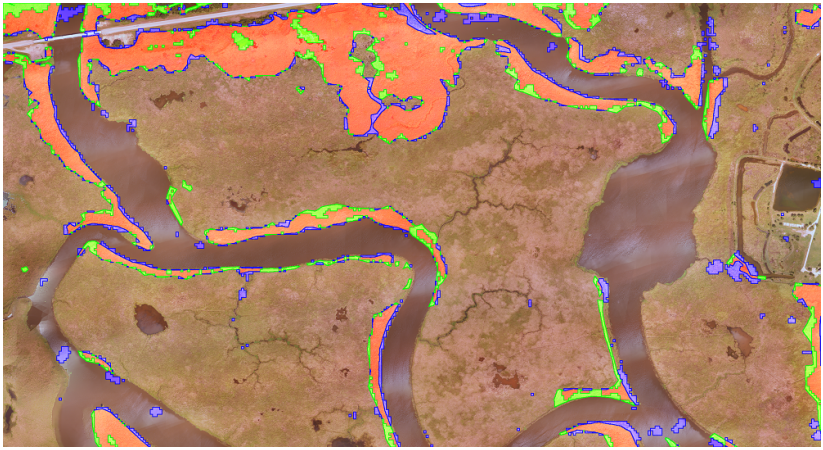
Other aquatic vegetation are a specific issue for the GLCM-RBFN method, being one of the main sources of degradation of the result. The low number of centers in the hidden layer may have directly influenced this aspect. This was the source of the error that caused the biggest accuracy difference between the maps. More results can be found on the annex A.

Figure 28 – Site I assessment for PNN classification using GLCM texture features.



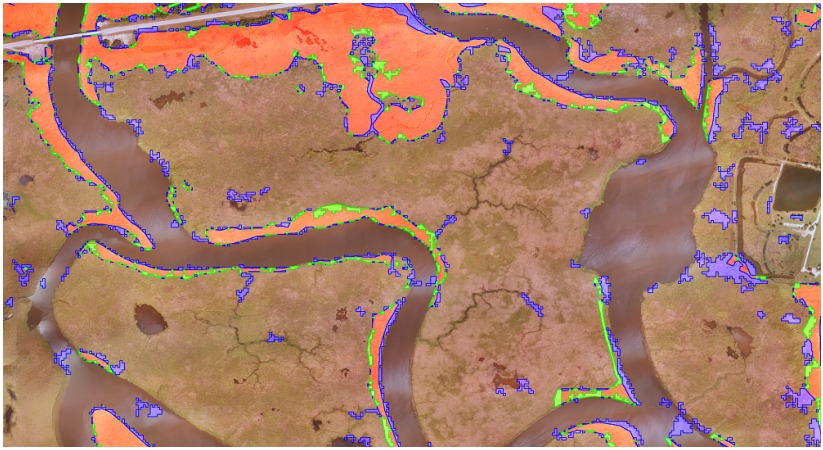
Source: Image from authors.

Figure 29 – Site I assessment for MLP classification using GLCM texture features.



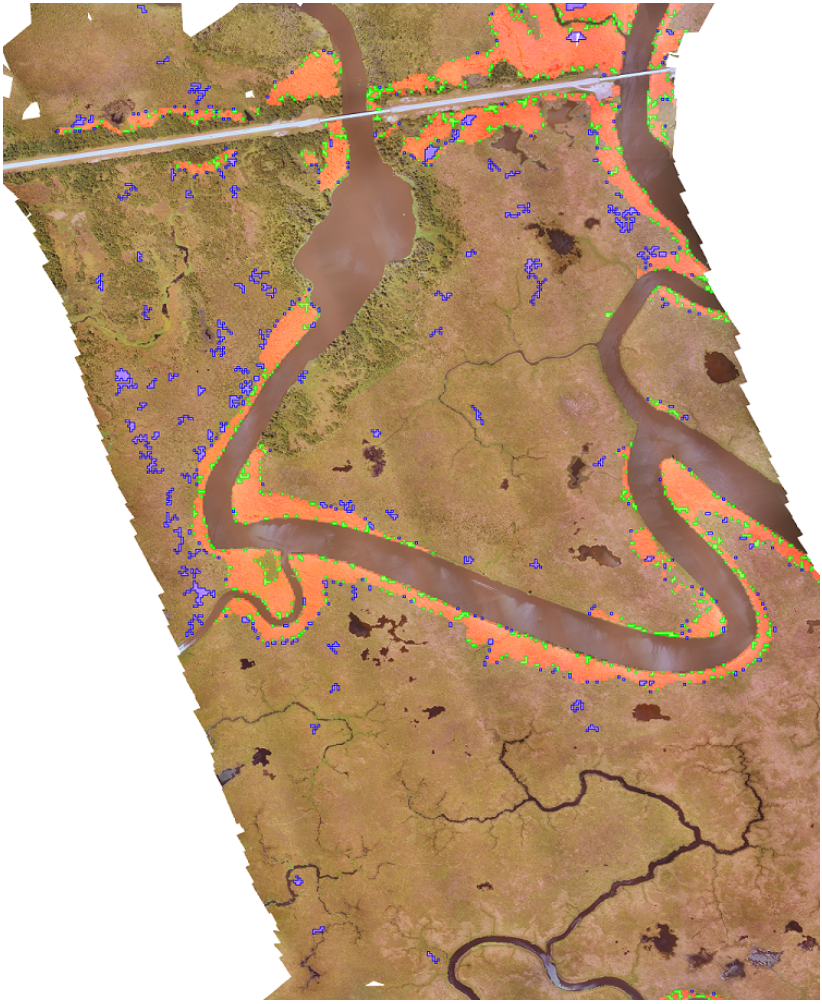
Source: Image from authors.

Figure 30 – Site I assessment for RBFN classification using GLCM texture feature.



Source: Image from authors.

Figure 31 – Site II assessment for RBFN classification using GLCM texture features.



Source: Image from authors.

6.2.2 Gabor Filters

Table 14 shows the statistical results for site I. The combination GF-PNN presented the best solution. Despite the specific poor agreement presented by the GF-RBFN solution, GF-PNN and GF-MLP solutions presented substantial agreement.

Table 14 – Site I - Texture Features Based on GF.

	PNN		MLP		RBFN		SVM	
Class	P	NP	P	NP	P	NP	P	NP
OE (%)	30.3	2.4	28.5	3.2	100	0	26.0	3.3
CE (%)	17.2	4.9	21.0	4.7	0	14.4	21.0	4.0
A (%)	69.7	97.6	71.5	96.8	0	100	73.0	96.6
κ	0.7200		0.7111		0.0		0.7100	
OA (%)	93.6		93.2		85.6		93.0	

Source: From authors.

Table 15 – Site II - Texture Features Based on GF.

	PNN		MLP		RBFN		SVM	
Class	P	NP	P	NP	P	NP	P	NP
OA (%)	13.5	1.4	28.6	3.8	100	0,0	28.4	2.5
CE (%)	17.0	1.1	40.7	2.2	0	0.07	31.0	2.2
A (%)	86.5	98.6	71.4	96.2	0	100	71.6	97.5
κ	0.8351		0.6177		0.0000		0.6700	
OA (%)	97.7		94.4		92.8		95.6	

Source: From authors.

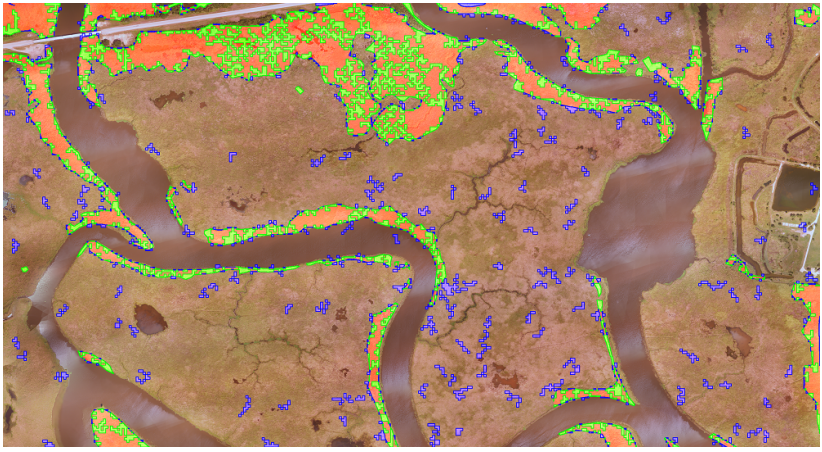
The GF based results for Site II are shown in Table 15. As well as for Site I, the GF-RBFN did not converge again independently of the parameters used in the hyperparameter tuning step. Two linked factors can be associated to this problem: First, the number of centers may not have been enough to differentiate the objects. That is, as demonstrated in the GF-RBFN combination, a larger number of neurons in the hidden layer may have helped in the final result. Second, as shown in the Figure 14, the encoding of the main blocks is very similar. This fact helps to explain the necessity of a higher number of centers to distinguish the

objects. The κ value for the GF-PNN solution is the unique in in the almost perfect agreement. State-of-art and GF-MLP solutions κ value are in the substantial agreement.

Although GF-PNN and GF-MLP combinations had a higher κ result if compared to the state-of-art for Site I, the same is not true for other parameters. As shown in Samiappan et al. (2016a) work, the *Phragmites* agreement is higher and omission error is lower. In addition, the difference in kappa value between the GF-MLP and state-of-art is practically negligible. A negative aspect presented by the best solution for Site I is the low *Phragmites* agreement.

The results presented for Site II changed drastically if compared to Site I results. All parameters calculated from the GF-PNN solution improved. In addition, the GF-MLP combination results decrease considerably. One of the main consequences is the missclassification of areas with low density of *Phragmites*.

Figure 32 – Site I assessment for PNN classification using texture features based on GF.

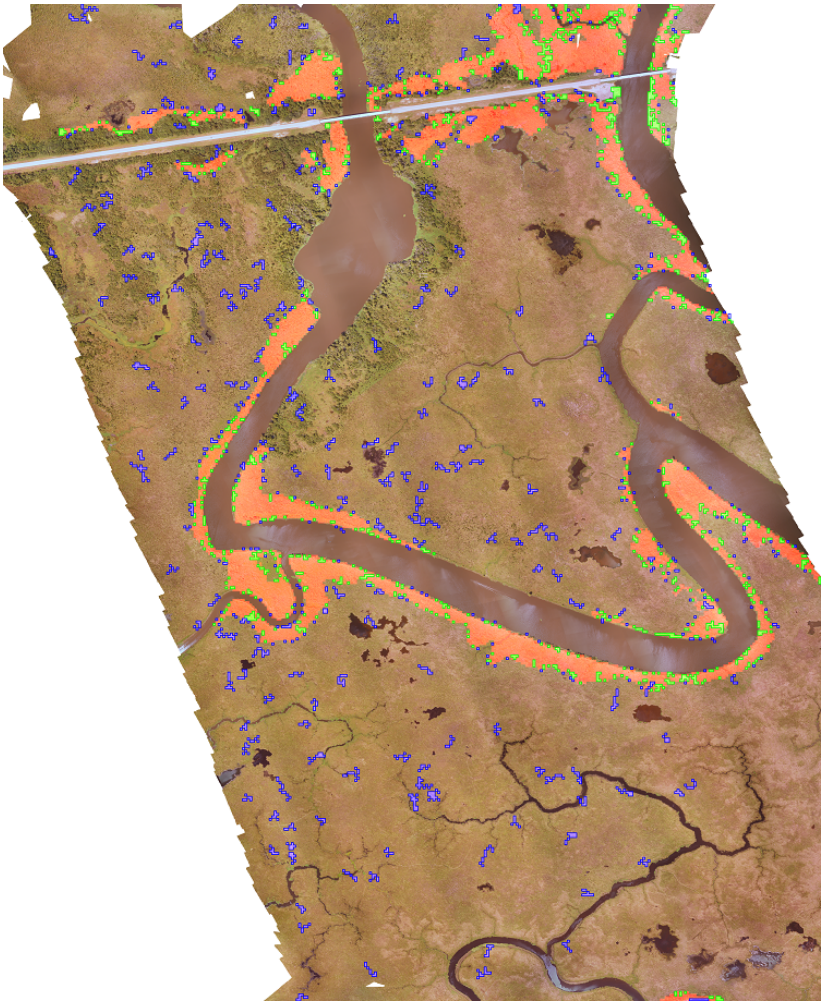


Source: Image from authors.

Furthermore to the problems reported in the GLCM results section, a negative aspect presented by all solutions based on GF is the high number of small stands that were missclassified (see Figures 32, 42, 33, and , 47 for more details). This issue was the main source of errors for the GF-MLP and it is an important problem for the re-

source managers, since it would cause additional effort visiting false positive stands. Different vegetation are the main responsible for these missclassified areas.

Figure 33 – Site I assessment for PNN classification using texture features based on GF.



Source: Image from authors.

6.2.3 Segmentation-based Fractal Texture Analysis

For the Site I, all combinations presented a value in the substantial or almost perfect agreement range (see Table 16).In this case, SFTA-PNN and SFTA-MLP have κ values above 0.8, where the first combination presented a very high value. The state-of-art and SFTA-MLP are in the substantial agreement range.

Table 16 – Site I - SFTA Texture Features.

	PNN		MLP		RBFN		SVM	
Class	P	NP	P	NP	P	NP	P	NP
OE (%)	11.9	1.31	16.4	2.0	30.3	2.4	15.8	5.8
CE (%)	8.2	2.0	12.4	2.7	17.2	4.9	29.5	2.7
A (%)	88.1	98.7	83.6	98.0	69.7	97.6	84.1	94.1
κ	0.8827		0.8319		0.7200		0.7200	
OA (%)	97.2		95.9		93.6		92.0	

Source: From authors.

As presented in the table 17, the SFTA-RBFN solution was the unique combination that did not present a κ above 0.61. As well as in the GF based solution, the SFTA-RBFN solution resulted in low *Phragmites* agreement in both Sites.

Table 17 – Site I - SFTA Texture Features.

	PNN		MLP		RBFN		SVM	
Class	P	NP	P	NP	P	NP	P	NP
OA (%)	10.0	0.1	23.4	0.9	68.8	0.3	34.1	2.1
CE (%)	1.5	0.8	13.5	1.8	12.2	5.1	28.7	2.6
A (%)	90.0	99.9	76.5	99.1	31.2	99.7	65.9	97.9
κ	0.9362		0.7985		0.4394		0.6613	
OA (%)	99.2		94.4		92.8		95.6	

Source: From authors.

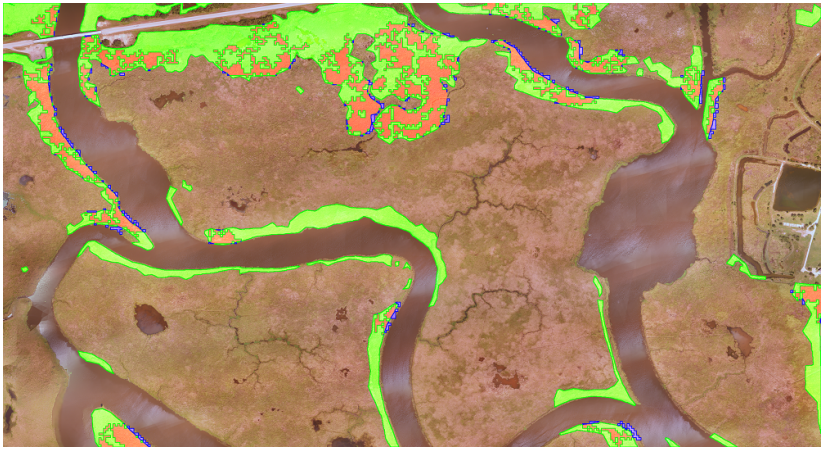
Unlike in the other feature extraction algorithms, this presented a clear pattern between the validation data in the Table 16 and 17. The SFTA-PNN resulted in the most reliable map in both Sites. In addition, SFTA-PNN and SFTA-MLP had similar agreement value for both

classes. The considerable drop in the *Phragmites* agreement value for SFTA-RBFN was the main anomaly. This fact may be a consequence of the low maximum number of centers defined for this work. As shown in the Figures 34 and 35, there are small stands identified as *Phragmites* within the real stand. This can be a consequence of a specific pattern identified by a center. In addition, major pattern change in the stands may have caused this smaller value. Despite these facts, the other solutions presented reasonable values of κ for the Site II.

The same problems described in the GLCM assessment section can be used to explain the sources of missclassification in the SFTA-PNN and SFTA-MLP solutions. As can be seen in the Figure 44, the problem related to the river sunlight reflection happened just with the SFTA-MLP solution. Part of the SFTA features come from the boundary complexity. Considering that and the blocks specificity, the values extracted for the boundaries are very similar. This difference may be not enough in the process of classification.

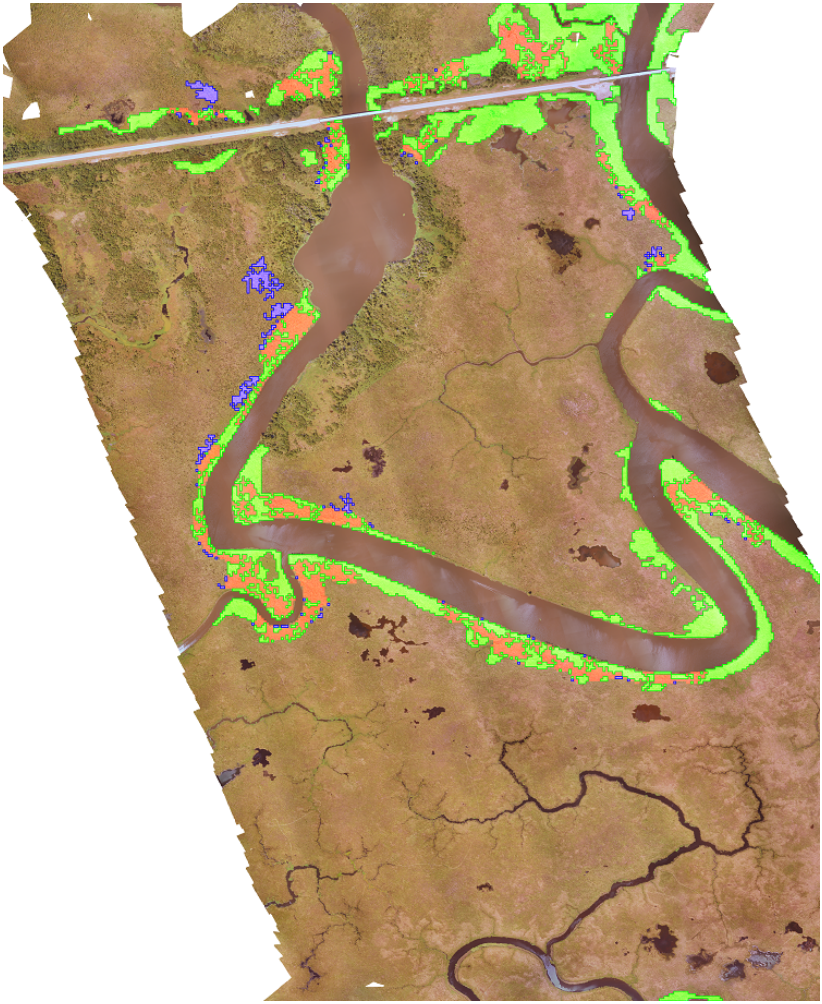
Regardless the reported problems, it is important to highlight the high capacity demonstrate by the SFTA-PNN solution (See Figure 36). In this combination, the main problems are related to the transition between P and NP. However, the system presented a good mapping capacity mainly in dense areas.

Figure 34 – Site I assessment for RBFN classification using SFTA texture features.



Source: Image from authors.

Figure 35 – Site II assessment for RBFN classification using SFTA.



Source: Image from authors.

Figure 36 – Site II assessment for PNN classification using SFTA texture features.

Source: Image from authors.

6.2.4 Wavelet Texture Analysis

Table 18 and 19 shows the statistical results obtained for the solutions based on WTA. In addition, they present the results obtained by Samiappan et al. (2016a) and Casagrande et al. (2017). In both situations, only the solutions based on WTA-PNN achieved a κ value above 0.8. The other results are between 0.61 and 0.8.

Table 18 – Site I - WTA. PNN-2 column represents Casagrande et al. (2017) results.

	PNN		MLP		RBFN		SVM		PNN-2	
Class	P	NP	P	NP	P	NP	P	NP	P	NP
OA (%)	8.4	1.1	19.8	4.0	18.9	3.0	27.0	2.6	14.8	0.4
CE (%)	6.8	1.4	24.1	3.1	18.0	3.2	17.9	4.3	3.1	2.4
A (%)	91.6	98.9	80.2	96.0	81.1	97.0	72.9	97.3	85.2	99.6
κ	0.9113		0.7443		0.7843		0.7300		0.8900	
OA (%)	97.8		93.9		94.7		93.0		97.5	

Source: From authors.

Table 19 – Site II - WTA. PNN-2 column represents Casagrande et al. (2017) results.

	PNN		MLP		RBFN		SVM		PNN-2	
Class	P	NP	P	NP	P	NP	P	NP	P	NP
OA (%)	10.8	0.1	30.3	0.9	29.6	1.3	35.6	1.3	20.6	0.2
CE (%)	1.3	0.8	14.9	2.3	18.9	2.3	16.5	3.6	3.8	1.6
A (%)	89.2	99.9	69.7	99.1	70.4	98.7	64.4	98.7	79.3	99.8
κ	0.9325		0.7500		0.7359		0.70		0.8600	
OA (%)	99.1		96.9		96.7		95.5		98.3	

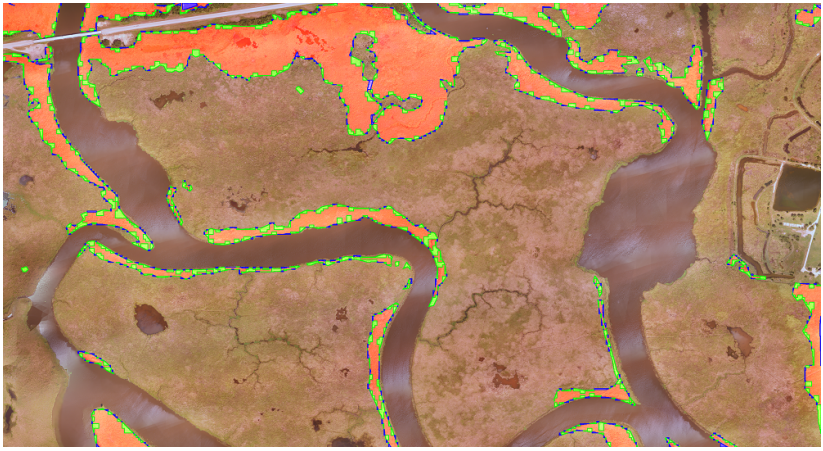
Source: From authors.

As well as in the GLCM-based solution, the WTA-RBFN had a good performance in both Sites. Specific aquatic and non-aquatic vegetations were the main source of error on the final map (Figures 45 and 51). In addition, low density *Phragmites* blocks contributed to the reduction of system accuracy.

As demonstrated in the Figures 38 and 49, the WTA-RBFN solution also had problems with aquatic plants. In addition, an anomaly

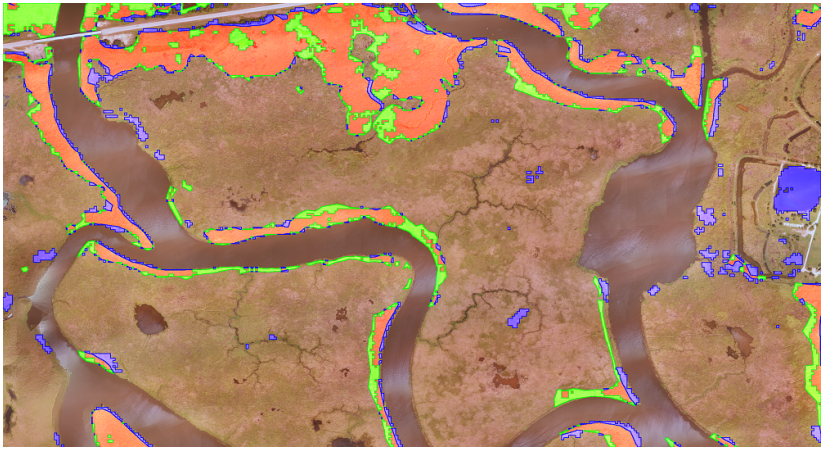
can be seen in Figure 38, where a lake was completely identified as *Phragmites*. WTA-PNN combination presented the most reliable map.

Figure 37 – Site I assessment for PNN classification using WTA.



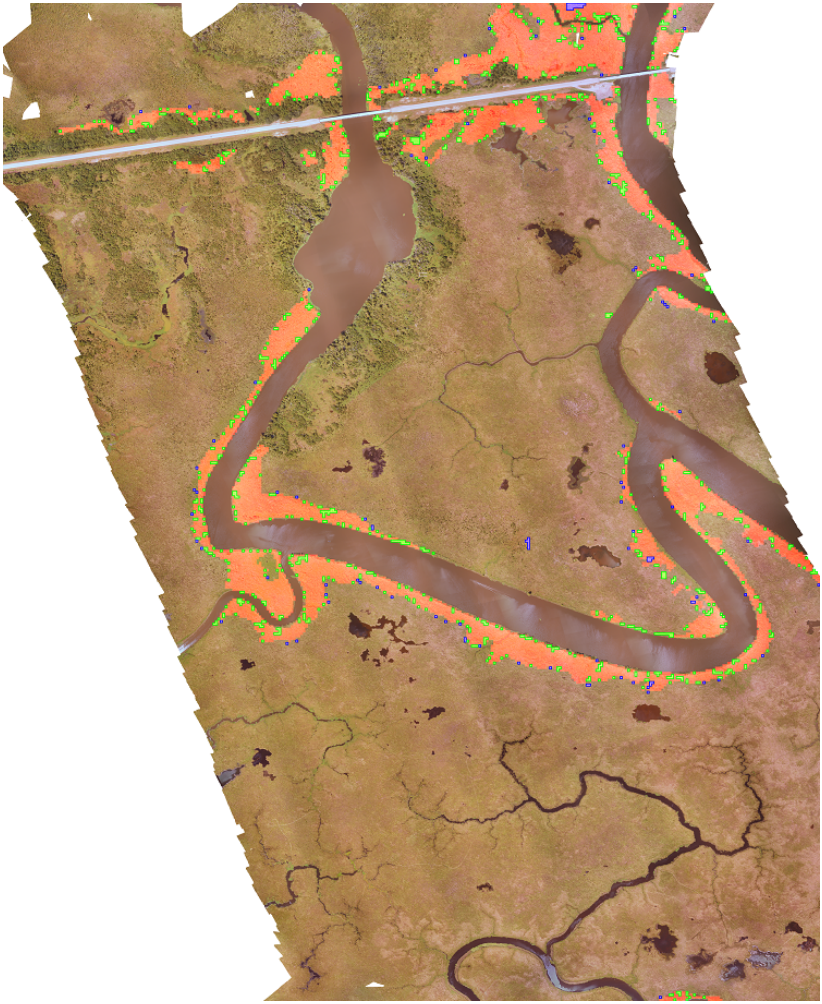
Source: Image from authors.

Figure 38 – Site I assessment for MLP classification using WTA.



Source: Image from authors.

Figure 39 – Site II assessment for PNN classification using WTA.



Source: Image from authors.

6.3 COMPARATIVE STUDY WITH OTHER TECHNIQUES

Based on the work by Kotsiantis (KOTSIANTIS, 2007), we performed a comparative study with other classification approaches. Since

decision trees can be translated into a set of rules by creating a separate rule for each path (QUINLAN, 1993), this study did not consider this kind of logical learning method. In addition, three ANN architectures were proposed in this work for mapping *Phragmites*, we did not include other neural networks for comparison.

Hence, our comparative study comprises the following classifiers: SVM, k-NN, Decision Tree (DT), Naive Bayes (NB), and the best results presented in the previous section for Site I and II. Tables 20 and 21 show the system's performance considering these classifiers for site I and II, respectively. The results for SVM were obtained from the work by Samiappan et al. (SAMIAPPAN et al., 2016a).

Table 20 – Learning algorithms assessments for Site I

	Best Result		SVM		K-NN		DT		NB	
Class	P	NP	P	NP	P	NP	P	NP	P	NP
OA (%)	8.4	1.1	27.0	2.6	7.7	7.4	12.8	6.7	10.4	11.1
CE (%)	6.8	1.4	17.9	4.3	32.3	1.4	31.4	2.3	42.5	1.9
A (%)	91.6	98.9	72.9	97.3	92.3	92.6	87.2	93.3	89.6	88.9
κ	0.9113		0.7300		0.7400		0.7200		0.6400	
OA (%)	97.8		93.0		92.6		92.4		89.0	

Source: From authors.

Table 21 – Learning algorithms assessments for Site II

	Best Result		SVM		K-NN		DT		NB	
Class	P	NP	P	NP	P	NP	P	NP	P	NP
OA (%)	10.0	0.1	35.6	1.3	10.0	7.7	12.4	9.5	24.6	9.7
CE (%)	1.5	0.8	16.6	3.6	45.5	1.1	51.4	1.4	55.6	2.7
A (%)	90.0	99.9	64.4	98.7	90.0	92.3	87.6	90.5	75.4	90.3
κ	0.9362		0.7000		0.7359		0.5700		0.5000	
OA (%)	99.2		95.5		92.1		90.2		89.0	

Source: From authors.

From the data presented in Tables 20 and 21, it is possible to conclude that there is an inverse relationship between *Phragmites* and not *Phragmites* agreement. While k-NN has a higher P agreement, it has a smaller NP agreement. In the other side, PNN and SVM have a higher NP agreement and a smaller P agreement. Despite this fact that PNN had not the highest *Phragmites* agreement, it had the highest not

Phragmites agreement, κ, and OA for Site I and II. k-NN presented the highest P agreement. Among the other learning algorithms, SVM had the highest OA for both Sites but k-NN had the best κ.

6.4 CPU X GPU ASSESSMENT

Only Site I was considered in this study. Table 22 shows the computational time necessary for the feature extraction method.

Table 22 – CPU Time

	GLCM	SFTA	WTA
Time	339.46s	349.11s	24.74s

Source: From authors.

As expected, WTA implementation took less computational time. This is the method that has the least number of iterations in its algorithm. In the other side, SFTA took the longest computational time. The same explanation can be used in this case, because SFTA has a considerable number of iterations in its algorithm mainly because the thresholds. Table 23 shows the computational time for these methods in the GPU.

Table 23 – GPU Time

	GLCM	WTA
Time	43.85s	25.82s

Source: From authors.

As in the CPU, the WTA algorithm took the shortest time. As it can be observed, the computational time required to run in both situations is practically the same. This fact is associated with the complexity of the algorithm and the associated time to copy the data into the device. The same is not true for the GLCM implementation. In this case, the GPU implementation obtained a speedup of 7.7 times.

Table 24 shows a comparison between classification methods with hyperparameter tuning.

Table 24 – CPU Hyperparameter Tuning and Map Generation steps

	GLCM	SFTA	WTA
PNN	709.8s	667.7s	535.1s
MLP	146008s	202425s	74266s
RBFN	48462.7s	78110.8s	34071.6s
SVM	355.1s	407.3s.	393.7s

Source: From authors.

Some important aspects can be concluded from these data. As expected the GA optimization takes much more time than the grid-search algorithm mainly by the number of combinations tested. Except by the SFTA-PNN combination, the solutions based on SFTA took more time than others. This can be explained by number of features. The same can be observed in the solutions based on WTA. Table 25 shows the computational time on GPU for the classifier that presented best accuracy.

Table 25 – GPU Hyperparameter Tuning and Map Generation steps for PNN solutions

	GLCM	SFTA	WTA
PNN	$18,5s$	$31,0s$	$4.7s$

Source: From authors.

A major speedup was found in the GPU implementation from PNN. As described in the subsection 5.4.3, this was already expected since this problem is very parallelizable. Beside the combinations, in the grid-search algorithm the instances were executed in parallel. That is, each block processed a combination and in each block the threads processed the instances. After, in the map generation step, the instances were divided into threads and blocks.

6.5 DISCUSSIONS

As discussed in the section 6.3, analyzing the statistical parameters values and maps presented in the section 6.2, it is possible to conclude that the same inverse relationship happened for the PNN, MLP, and RBFN based solutions. That is, there is a trade-off between

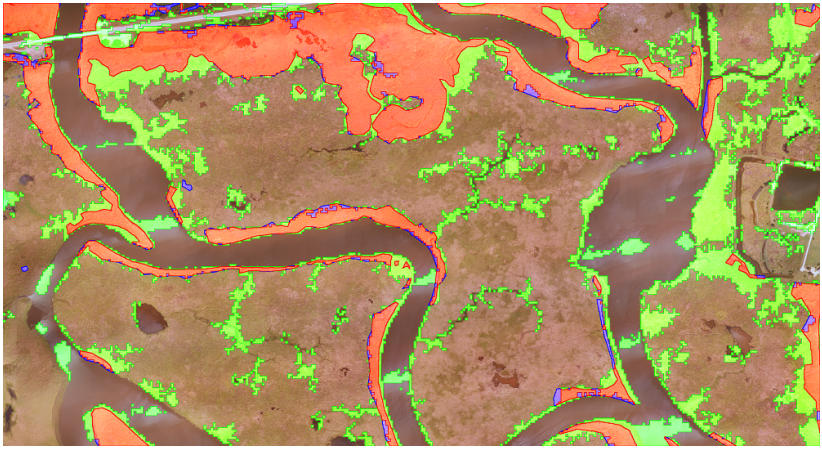
P and NP agreement. This is mainly caused by the proportion and the instances used in the training set of each classifier.

Among all problems discussed in the subsections from section 6.2 and in the section 6.3, the main source of errors are:

1. Transition between P and NP;
2. Intermixing of vegetations;
3. River sunlight reflection;

The first error can be found in all maps. This is a drawback from the block processing method. Although not so frequent in the maps presented, the intermixing of vegetations problem can be found in blocks where *Phragmites* mix with trees or aquatic vegetation. This kind of problem has already been reported in related works. The third problem happened only in the GLCM, GF, and SFTA based solutions. As described, a specific *Phragmites* characteristic was responsible for this missclassification. This problem could be solved including a pre-extraction step aiming to remove the river object.

Figure 40 – Site I assessment for k-NN classification, where *Phragmites* agreement is outlined in red, omission cases outlined in green, and commission cases in blue.



Source: Image from authors.

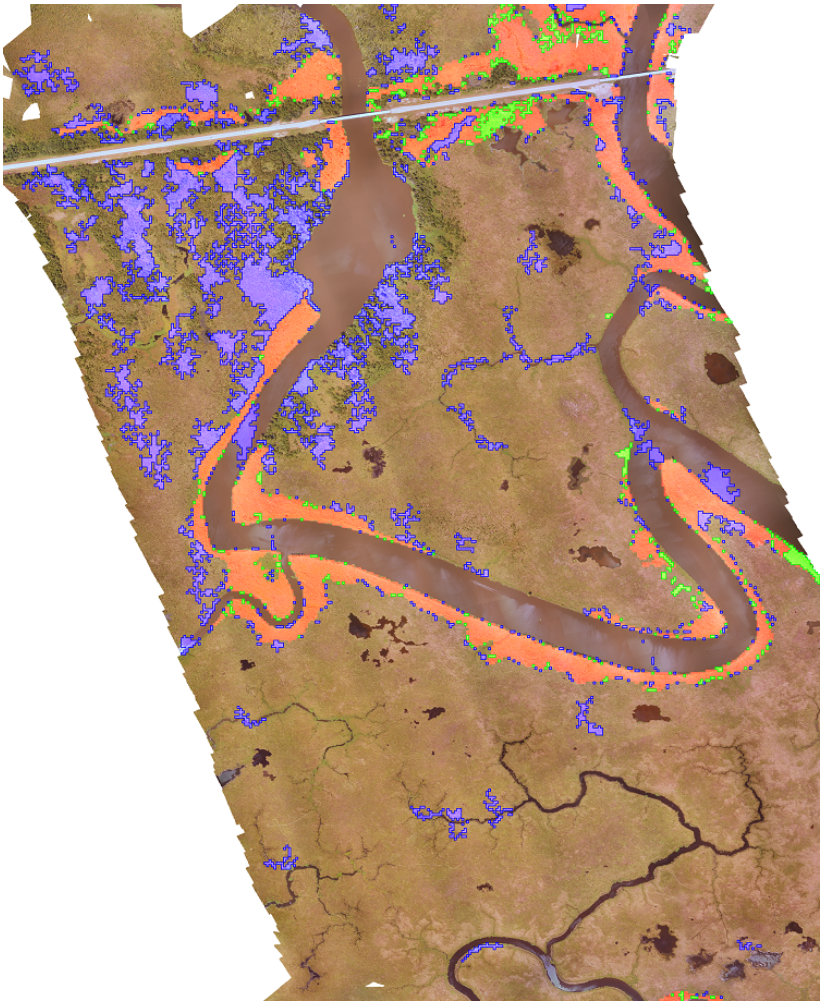
Although these problems, the best solution for Site I (WTA-PNN) and Site II (SFTA-PNN) presented maps with a high level of

reliability. If we compare the best solutions presented in the section 6.2 with the Samiappan et al. (2016a), and k-NN results (Figures 40 and 41), it is possible to verify that PNN was the unique technique that did not present new *Phragmites* patches that don't exist. That is, the system's misclassification were mainly located in the boundaries, and, in few cases, in the middle of real patches. This error remains a problem for remote monitoring efforts, but it means that the resource managers would not need to move to new areas that do not have *Phragmites*. This reaffirms the fact that, although there is still room for improvement, the proposed solution was able to considerably improve the confidence level of the generated maps. Although the best results from this work and Casagrande et al. (2017) are very similar, a major improvement in *Phragmites* accuracy was found mainly because the of an hyperparameter tuning step. This represents a major improvement for resource managers as this generates more accurate information that can be used to define the best solution for the area in analysis.

Another important aspect to highlight from section 6.3 is the PNN performance dominance over other techniques for all performed tests, except for the Site I GLCM-MLP solution. The same conclusion was made by other authors in (ZHAO; HUANG; GUO, 2003; MIRZAEI et al., 2011). Although it requires more memory than the other techniques, it demonstrated a high potential for the problem described in this work. Furthermore, SFTA and WTA also showed a higher potential when compared to GLCM and GF.

In the section 6.4 two important facts must be highlighted: First, the CPU implementation in C reduced considerably the necessary time per squared kilometer. In Samiappan et al. (2016a), the authors described that the overall processing time per square kilometre is approximately 50 min. Although in this case we are not considering the mosaicing step, the overall processing time for the same solution in C was approximately 12 minutes for the whole Site I. The same happened if we compare the WTA-PNN implementation proposed in this work with Casagrande et al. (2017) MATLAB implementation. In this case, the overall processing time was approximately 10 minutes for the whole Site I against 45 minutes per square kilometer. Second, the GPU implementation for this problem is viable and it greatly reduced the processing time required to generate a map. This improvement is essential for the resource managers, since time is an essential factor in *Phragmites* propagation. That is, the possibility of a map with high level of reliability being generated in few hours can an important monitoring tool for resource managers.

Figure 41 – Site II assessment for k-NN classification, where *Phragmites* agreement is outlined in red, omission cases outlined in green, and commission cases in blue.



Source: Image from authors.

7 CONCLUSION

In this work it was presented a comparative study of image texture analysis and machine learning methods for classification of *Phragmites australis* in imagery aquired by LARS with a small, hand-launched UAS. In addition, a GPU implementation for the best results was presented aiming to reduce the total computational time necessary to generate the map.

For comparison and consistency, it was performed evaluation with the same imagery aquired by Samiappan et al. (2016a). The experimental study presented in this work shows that the combination of PNN with WTA for Site I and PNN with SFTA for Site II yields superior classification accuracies if compared to the other combinations proposed here. In addition, the same was found when compared to the state-of-art results. A major improvement in *Phragmites* agreement for both Sites resulted in a higher overall accuracy and κ value.

Based on experimental results presented in this research, WTA and SFTA presented the best capacity to distinguish *Phragmites* stands from other objects accurately. In addition, PNN-based solutions had the best results in the majority of the cases. This fact was already expected, since the same conclusion was found in works that compared the same ANN architectures.

Furthermore, the necessary computational cost necessary to generate the final map was reduced considerably in this work through a GPU implementation. This result was already expected too, since the methodology proposed is very parallelizable. In addition, the blocks and the instances do not have a direct relation, reducing the implementation complexity of the procedures.

As future work, we propose to parallelize the other methods of classification with a GPGPU. In addition, aiming to reduce the *Phragmites* omission error, we propose the implementation of recursive block subdivision and further classification when the original block summation is below a specific threshold.

REFERENCES

ADAM, E.; MUTANGA, O.; RUGEGE, D. Multispectral and hyperspectral remote sensing for identification and mapping of wetland vegetation: a review. *Wetlands Ecology and Management*, Springer Nature, v. 18, n. 3, p. 281–296, dec 2009.

AGGARWAL, S. Principles of remote sensing. In: *Satellite Remote Sensing and GIS Applications in Agricultural Meteorology*. [S.l.: s.n.], 2003. p. 23–38.

AHARKAVA, L. *Artificial neural networks and self-organization for knowledge extraction*. Dissertação (Mestrado) — Charles University in Prague, 2010.

AKANDE, K. O. et al. Performance comparison of SVM and ANN in predicting compressive strength of concrete. *IOSR Journal of Computer Engineering*, IOSR Journals, v. 16, n. 5, p. 88–94, 2014.

ALTAVIAN. *Altavian Nova UAS*. 2017. <https://www.altavian.com/product/nova/>.

ANDERSON, K.; GASTON, K. J. Lightweight unmanned aerial vehicles will revolutionize spatial ecology. *Frontiers in Ecology and the Environment*, Wiley-Blackwell, v. 11, n. 3, p. 138–146, apr 2013.

ANTKOWIAK, M. *Artificial Neural Networks vs. Support Vector Machines for Skin Diseases Recognition*. Dissertação (Mestrado) — Umea University, 2006.

ARTIGAS, F.; PECHMANN, I. C. Balloon imagery verification of remotely sensed phragmites australis expansion in an urban estuary of new jersey, usa. *Landscape and Urban Planning*, v. 95, n. 3, p. 105 – 112, 2010. ISSN 0169-2046.

AYAT, S. et al. A comparison of artificial neural networks learning algorithms in predicting tendency for suicide. *Neural Computing and Applications*, Springer Nature, v. 23, n. 5, p. 1381–1386, Jul 2012.

BECKER, B. Mapping invasive plants in wetlands. *SPIE Newsroom*, SPIE-Intl Soc Optical Eng, 2009.

BELAKHDAR, I. et al. A comparison between ANN and SVM classifier for drowsiness detection based on single EEG channel. In: *2016 2nd International Conference on Advanced Technologies for Signal and Image Processing (ATSIP)*. [S.l.]: IEEE, 2016.

BENARDOS, P.; VOSNIAKOS, G.-C. Optimizing feedforward artificial neural network architecture. *Engineering Applications of Artificial Intelligence*, Elsevier BV, v. 20, n. 3, p. 365–382, apr 2007.

BERTNESS, M. D.; EWANCHUK, P. J.; SILLIMAN, B. R. Anthropogenic modification of new england salt marsh landscapes. *Proceedings of the National Academy of Sciences*, Proceedings of the National Academy of Sciences, v. 99, n. 3, p. 1395–1398, jan 2002.

BHARATI, M. H.; LIU, J.; MACGREGOR, J. F. Image texture analysis: methods and comparisons. *Chemometrics and Intelligent Laboratory Systems*, Elsevier BV, v. 72, n. 1, p. 57–71, jun 2004.

BIRD, D.; KYLE, A. S. *Conservation Area Plan For the Pearl River*. CFMS Cooperative Agreement No. 583066, August 2004. 59 p.

BOURGEAU-CHAVEZ, L. L. et al. Mapping invasive phragmites australis in the coastal great lakes with ALOS PALSAR satellite imagery for decision support. *Journal of Great Lakes Research*, Elsevier BV, v. 39, p. 65–77, 2013.

BRANDSTETTER, A.; ARTUSI, A. Radial basis function networks gpu-based implementation. *IEEE Transactions on Neural Networks*, v. 19, n. 12, p. 2150–2154, Dec 2008. ISSN 1045-9227.

BROOKS, H. *Exotic Vegetation Assesment (EVA): Remote Sensing Phragmites Australis at Presque Isle State Park*. Dissertação (Mestrado) — The Pennsylvania State University Schreyer Honors College, 2014.

BROOMHEAD, D.; LOWE, D. Multivariable functional interpolation and adaptive networks. *Complex Systems*, v. 2, p. 321–355, 1988.

BURGES, C. J. A tutorial on support vector machines for pattern recognition. *Data mining and knowledge discovery*, Springer, v. 2, n. 2, p. 121–167, 1998.

CAMPBELL, J. B. *Introduction to Remote Sensing: Second Edition*. [S.l.]: The Guilford Press, 1996. ISBN 1572300418.

CARPENTER, A. C. A. *cuSVM: a CUDA implementation of support vector classification and regression.* jan 2009. <http://patternsonascreen.net/cuSVMDesc.pdf>.

CASAGRANDE, L. et al. Probabilistic neural network and wavelet transform for mapping of phragmites australis using low altitude remote sensing. In: *2017 30th SIBGRAPI Conference on Graphics, Patterns and Images (SIBGRAPI).* [S.l.: s.n.], 2017. p. 269–276.

CHAMBERS, R. M.; MEYERSON, L. A.; DIBBLE, K. L. Ecology of phragmites australis and responses to tidal restoration. In: *Tidal Marsh Restoration.* [S.l.]: Island Press/Center for Resource Economics, 2012. p. 81–96.

CHANG, C.-C.; LIN, C.-J. Libsvm: A library for support vector machines. *ACM Trans. Intell. Syst. Technol.*, ACM, New York, NY, USA, v. 2, n. 3, p. 27:1–27:27, maio 2011. ISSN 2157-6904.

CHE, S. et al. A performance study of general-purpose applications on graphics processors using CUDA. *Journal of Parallel and Distributed Computing*, Elsevier BV, v. 68, n. 10, p. 1370–1380, oct 2008.

COLWELL, R. N. Determining the prevalence of certain cereal crop diseases by means of aerial photography. *Hilgardia*, University of California Agriculture and Natural Resources (UC ANR), v. 26, n. 5, p. 223–286, nov 1956.

COSTA, A. F.; HUMPIRE-MAMANI, G.; TRAINA, A. J. M. An efficient algorithm for fractal analysis of textures. In: *2012 25th SIBGRAPI Conference on Graphics, Patterns and Images.* [S.l.: s.n.], 2012. p. 39–46. ISSN 1530-1834.

COURTLAND, R. *Transistors Could Stop Shrinking in 2021.* 2016. <https://spectrum.ieee.org/semiconductors/devices/transistors-could-stop-shrinking-in-2021>.

CRONK, J. K.; FENNESSY, M. S. *Wetland Plants: Biology and Ecology.* [S.l.]: CRC Press, 2001. ISBN 1566703727.

DAUGMAN, J. G. Uncertainty relation for resolution in space, spatial frequency, and orientation optimized by two-dimensional visual cortical filters. *Journal of the Optical Society of America*, v. 2, p. 1160–1169, 1985.

DING, C. *CUDA Tutorial*. 2017. *<http : //geco.mines.edu/tesla/cuda_tutorial_mio/>*.

DUNN, D.; HIGGINS, W. E. Optimal gabor filters for texture segmentation. *IEEE Transactions on Image Processing*, v. 4, n. 7, p. 947–964, Jul 1995. ISSN 1057-7149.

DUTT, V.; CHAUDHRY, V.; KHAN, I. Pattern recognition: an overview. *American Journal of Intelligent Systems*, Scientific and Academic Publishing, v. 2, n. 1, p. 23–27, aug 2012.

DYKIN, V. S. et al. Application of genetic algorithm to configure artificial neural network for processing a vector multisensor array signal. In: *2015 International Siberian Conference on Control and Communications (SIBCON)*. [S.l.]: Institute of Electrical and Electronics Engineers (IEEE), 2015.

EASTON, R. L. *Fundamentals of Digital Image Processing*. nov 2017.

ELIE, S. An overview of pattern recognition. *Antonine University Article*, 2013. University of Burgundy.

FRANCO, J. et al. A parallel implementation of the 2d wavelet transform using CUDA. In: *2009 17th Euromicro International Conference on Parallel, Distributed and Network-based Processing*. [S.l.]: IEEE, 2009.

GABOR, D. Theory of communication. part 1: The analysis of information. *Journal of the Institution of Electrical Engineers - Part III: Radio and Communication Engineering*, Institution of Engineering and Technology (IET), v. 93, n. 26, p. 429–441, nov 1946.

GANATRA, A. et al. Initial classification through back propagation in a neural network following optimization through GA to evaluate the fitness of an algorithm. *International Journal of Computer Science and Information Technology*, Academy and Industry Research Collaboration Center (AIRCC), v. 3, n. 1, p. 98–116, feb 2011.

GARLAND, M. et al. Parallel computing experiences with cuda. *IEEE Micro*, v. 28, n. 4, p. 13–27, July 2008. ISSN 0272-1732.

GARRO, B. A.; VÁZQUEZ, R. A. Designing artificial neural networks using particle swarm optimization algorithms. *Computational Intelligence and Neuroscience*, Hindawi Publishing Corporation, v. 2015, p. 1–20, 2015.

GILMORE, M. S. et al. Integrating multi-temporal spectral and structural information to map wetland vegetation in a lower connecticut river tidal marsh. *Remote Sensing of Environment*, Elsevier BV, v. 112, n. 11, p. 4048–4060, nov 2008.

GOMAN, M.; WELLS, L. Trends in river flow affecting the northeastern reach of the san francisco bay estuary over the past 7000 years. *Quaternary Research*, Cambridge University Press (CUP), v. 54, n. 02, p. 206–217, sep 2000.

GOMEZ, W.; PEREIRA, W. C. A.; INFANTOSI, A. F. C. Analysis of co-occurrence texture statistics as a function of gray-level quantization for classifying breast ultrasound. *IEEE Transactions on Medical Imaging*, v. 31, n. 10, p. 1889–1899, Oct 2012. ISSN 0278-0062.

GREFENSTETTE, J. Optimization of control parameters for genetic algorithms. *IEEE Transactions on Systems, Man, and Cybernetics*, Institute of Electrical and Electronics Engineers (IEEE), v. 16, n. 1, p. 122–128, jan 1986.

HAGAN, M. T.; MENHAJ, M. B. Training feedforward networks with the marquardt algorithm. *IEEE Transactions on Neural Networks*, v. 5, n. 6, p. 989–993, Nov 1994. ISSN 1045-9227.

HAGHIGHAT, M.; ZONOUZ, S.; ABDEL-MOTTALEB, M. Identification using encrypted biometrics. In: *Computer Analysis of Images and Patterns*. [S.l.]: Springer Berlin Heidelberg, 2013. p. 440–448.

HARALICK, R. M.; SHANMUGAM, K.; DINSTEIN, I. Textural features for image classification. *IEEE Transactions on Systems, Man, and Cybernetics*, Institute of Electrical and Electronics Engineers (IEEE), SMC-3, n. 6, p. 610–621, nov 1973.

HAYKIN, S. *Redes Neurais - Principios E Prática (Em Portuguese do Brasil)*. [S.l.]: Bookman, 2003. ISBN 8573077182.

HAYKIN, S. *Redes Neurais Princípios e Práticas*. [S.l.]: Bookman, 2007.

HAYKIN, S. O. *Neural Networks and Learning Machines*. [S.l.]: Pearson, 2011.

HAZELTON, E. L. G. et al. Phragmites australis management in the united states: 40 years of methods and outcomes. *AoB PLANTS*, Oxford University Press (OUP), v. 6, n. 0, p. plu001–plu001, jan 2014.

HEBB, D. *The Organization of Behavior: A Neuropsychological Theory*. [S.l.]: Psychology Press, 1949. ISBN 0805843000.

HONG, H.; ZHENG, L.; PAN, S. Computation of gray-level co-occurrence matrix based on CUDA and its optimization. *CoRR*, abs/1710.06189, 2017.

HSU, C.-W.; CHANG, C.-C.; LIN, C.-J. *A Practical Guide to Support Vector Classification*. [S.l.], 2003.

HSU, C.-W. et al. A practical guide to support vector classification. 2003.

HUSSON, E.; ECKE, F.; REESE, H. Comparison of manual mapping and automated object-based image analysis of non-submerged aquatic vegetation from very-high-resolution UAS images. *Remote Sensing*, MDPI AG, v. 8, n. 9, p. 724, sep 2016.

JALI, M. H. et al. Pattern recognition of emg signal during load lifting using artificial neural network (ann). In: *2015 IEEE International Conference on Control System, Computing and Engineering (ICCSCE)*. [S.l.: s.n.], 2015. p. 172–177.

JANSSEN, L.; VANDERWEL, F. Accuracy assessment of satellite derived land-cover data: A review. *Photogrammetric Engineering and Remote Sensing; (United States)*, v. 60:4, Apr 1994.

JANSSEN, L. L. F.; HUURNEMAN, G. C. *Principles of Remote Sensing - An introductory textbook*. [S.l.]: ITC Educational Textbooks, 2001.

JAROS, J.; POSPICHAL, P. A fair comparison of modern CPUs and GPUs running the genetic algorithm under the knapsack benchmark. In: *Applications of Evolutionary Computation*. [S.l.]: Springer Berlin Heidelberg, 2012. p. 426–435.

JR., C. L. N. *Inteligência Artificial em Controle e Automação*. [S.l.]: Edgard Blucher, 2004. ISBN 8521203101.

JR, C. T. et al. Fast feature selection using fractal dimension. *Journal of Information and Data Management*, v. 1, n. 1, p. 3–16, 2010.

JäHNE, B. *Digital Image Processing*. [S.l.]: Springer, 1997. ISBN 3540240357.

KETTENRING, K. M. et al. Phragmites australis (common reed) invasion in the rhode river subestuary of the chesapeake bay: Disentangling the effects of foliar nutrients, genetic diversity, patch size, and seed viability. *Estuaries and Coasts*, Springer Nature, v. 33, n. 1, p. 118–126, dec 2009.

KOTSIANTIS, S. B. Supervised machine learning: A review of classification techniques. In: *Proceedings of the 2007 Conference on Emerging Artificial Intelligence Applications in Computer Engineering: Real Word AI Systems with Applications in eHealth, HCI, Information Retrieval and Pervasive Technologies*. Amsterdam, The Netherlands, The Netherlands: IOS Press, 2007. p. 3–24. ISBN 978-1-58603-780-2.

KPALMA, K.; RONSI, J. An overview of advances of pattern recognition systems in computer vision. In: *Vision Systems: Segmentation and Pattern Recognition*. [S.l.]: I-Tech Education and Publishing, 2007.

LAMB, D.; BROWN, R. PA—precision agriculture. *Journal of Agricultural Engineering Research*, Elsevier BV, v. 78, n. 2, p. 117–125, feb 2001.

LANTZ, N. J.; WANG, J. Object-based classification of worldview-2 imagery for mapping invasive common reed, phragmites australis. *Canadian Journal of Remote Sensing*, Taylor & Francis, v. 39, n. 4, p. 328–340, 2013.

LEVENBERG, K. A method for the solution of certain non-linear problems in least squares. *Quarterly of Applied Mathematics*, American Mathematical Society (AMS), v. 2, n. 2, p. 164–168, jul 1944.

LIAO, P.-S.; CHEN, T.-S.; CHUNG, P.-C. A fast algorithm for multilevel thresholding. *JOURNAL OF INFORMATION SCIENCE AND ENGINEERING*, p. 713–727, 2001.

LIN, S.-W. et al. Particle swarm optimization for parameter determination and feature selection of support vector machines. *Expert systems with applications*, Elsevier, v. 35, n. 4, p. 1817–1824, 2008.

LIVENS, S. et al. Wavelets for texture analysis, an overview. In: *1997 Sixth International Conference on Image Processing and Its Applications*. [S.l.: s.n.], 1997. v. 2, p. 581–585 vol.2. ISSN 0537-9989.

LIWA, E. J. *A Neural Network Model for Classification of Coastal Wetlands Vegetation Structure with Moderate Resolution Imaging Spectro-Radiometer (MODIS) Data*. Tese (Doutorado) — Louisiana State University, 2006.

M LAPIN B, R. J. M. Phragmites australis (p. communis): threats, management and monitoring. *NATURAL AREAS JOURNAL*, p. 285–294, 1994.

MALONEY, L. T.; WANDELL, B. A. Color constancy: a method for recovering surface spectral reflectance. *J. Opt. Soc. Am. A*, OSA, v. 3, n. 1, p. 29–33, Jan 1986.

MARCEAU, D. J. et al. Evaluation of the grey-level co-occurrence matrix method for land-cover classification using spot imagery. *IEEE Transactions on Geoscience and Remote Sensing*, v. 28, n. 4, p. 513–519, Jul 1990. ISSN 0196-2892.

MARQUARDT, D. W. An algorithm for least-squares estimation of nonlinear parameters. *Journal of the Society for Industrial and Applied Mathematics*, Society for Industrial & Applied Mathematics (SIAM), v. 11, n. 2, p. 431–441, jun 1963.

MARTIN, L. J.; BLOSSEY, B. The runaway weed: Costs and failures of phragmites australis management in the USA. *Estuaries and Coasts*, Springer Nature, v. 36, n. 3, p. 626–632, mar 2013.

MATUSZYK, P. et al. A comparative study on hyperparameter optimization for recommender systems. In: LEX, E. et al. (Ed.). *Workshop on Recommender Systems and Big Data Analytics (RS-BDA'16) @ iKNOW 2016*. [S.l.: s.n.], 2016.

MCCULLOCH, W. S.; PITTS, W. A logical calculus of the ideas immanent in nervous activity. *The bulletin of mathematical biophysics*, v. 5, n. 4, p. 115–133, Dec 1943. ISSN 1522-9602.

MEDAPATI, J. et al. Performance comparison of radial basis function networks and probabilistic neural networks for telugu character recognition. *Global Journal of Computer Science and Technology*, 2011.

MEYERSON, L. A.; CHAMBERS, R. M.; VOGT, K. A. The effects of phragmites removal on nutrient pools in a freshwater tidal marsh ecosystem. *Biological Invasions*, Springer Nature, v. 1, n. 2/3, p. 129–136, 1999.

MIRZAEI, M. et al. Comparative analysis of probabilistic neural network, radial basis function, and feed-forward neural network for fault classification in power distribution systems. *Electric Power Components and Systems*, Taylor & Francis, v. 39, n. 16, p. 1858–1871, 2011.

MOZDZER, T. J.; ZIEMAN, J. C. Ecophysiological differences between genetic lineages facilitate the invasion of non-nativePhragmites australisin north american atlantic coast wetlands. *Journal of Ecology*, Wiley-Blackwell, v. 98, n. 2, p. 451–458, mar 2010.

NASA. *Remote Sensing*. 2017. <https://earthobservatory.nasa.gov/Features/RemoteSensing/>.

NIERING, W. A.; WARREN, R. S.; WEYMOUTH, C. G. *Bulletin No. 22: Our Dynamic Tidal Marshes: Vegetation Changes as Revealed by Peat Analysis*. 1977. Bulletins.

NVIDIA. *WHAT IS CUDA?* 2017. <*http* : *//www.nvidia.com/object/cuda_home_new.html*>.

ORYCH, A. Review of Methods for Determining the Spatial Resolution of UAV Sensors. *ISPRS - International Archives of the Photogrammetry, Remote Sensing and Spatial Information Sciences*, Copernicus GmbH, XL-1/W4, p. 391–395, aug 2015. <https://doi.org/10.5194/isprsarchives-xl-1-w4-391-2015>.

PATEL, K.; VALA, J.; PANDYA, J. Comparison of Various Classification Algorithms on Iris Datasets Using Weka. *International Journal of Advance Engineering and Research Development*, International Journal of Advance Engineering and Research Development (IJAERD), v. 1, n. 1, p. 01–07, feb 2014.

PATER, L. Application of artificial neural networks and genetic algorithms for crude fractional distillation process modeling. *CoRR*, abs/1605.00097, 2016.

PENGRA, B. W.; JOHNSTON, C. A.; LOVELAND, T. R. Mapping an invasive plant, phragmites australis, in coastal wetlands using the EO-1 hyperion hyperspectral sensor. *Remote Sensing of Environment*, Elsevier BV, v. 108, n. 1, p. 74–81, may 2007.

PHAUDPHUT, C.; SO-IN, C.; PHUSOMSAI, W. A parallel probabilistic neural network ecg recognition architecture over gpu

platforms. In: *2016 13th International Joint Conference on Computer Science and Software Engineering (JCSSE)*. [S.l.: s.n.], 2016. p. 1–7.

PIX4D. *TOOLS - GSD Calculator*. nov 2017. <https://support.pix4d.com/hc/en-us/articles/202560249-TOOLS-GSD-Calculator>.

POWELL, M. J. D. Algorithms for approximation. In: MASON, J. C.; COX, M. G. (Ed.). New York, NY, USA: Clarendon Press, 1987. cap. Radial Basis Functions for Multivariable Interpolation: A Review, p. 143–167. ISBN 0-19-853612-7.

QUINLAN, J. R. *C4.5: Programs for Machine Learning*. San Francisco, CA, USA: Morgan Kaufmann Publishers Inc., 1993. ISBN 1-55860-238-0.

RAIZER, K. et al. Performance comparison of training methods for mlp neural networks in gpu. In: *VI CONGRESSO NACIONAL DE ENGENHARIA MECÂNICA*. [S.l.: s.n.], 2010. p. 1–9.

RAUBER, T. W. *Redes Neurais Artificiais*. 1998.

RIHA, L.; SMID, R. Acceleration of acoustic emission signal processing algorithms using CUDA standard. *Computer Standards & Interfaces*, Elsevier BV, v. 33, n. 4, p. 389–400, jun 2011.

ROSENBLATT, F. The perceptron: A probabilistic model for information storage and organization in the brain. *Psychological Review*, p. 65–386, 1958.

ROSENFELD, A.; WECHSLER, H. Pattern recognition: Historical perspective and future directions. *International Journal of Imaging Systems and Technology*, Wiley-Blackwell, v. 11, n. 2, p. 101–116, 2000.

SALTONSTALL, K. Cryptic invasion by a non-native genotype of the common reed, phragmites australis, into north america. *Proceedings of the National Academy of Sciences*, Proceedings of the National Academy of Sciences, v. 99, n. 4, p. 2445–2449, feb 2002.

SALTONSTALL, K. Genetic variation among north american populations ofPhragmites australis: Implications for management. *Estuaries*, Springer Nature, v. 26, n. 2, p. 444–451, apr 2003.

SALTONSTALL, K. et al. *Native and Non-native Phragmites*. [S.l.], 2005.

SAMIAPPAN, S. et al. Using unmanned aerial vehicles for high-resolution remote sensing to map invasivePhragmites australisin coastal wetlands. *International Journal of Remote Sensing*, Informa UK Limited, p. 1–19, oct 2016.

SAMIAPPAN, S. et al. Mapping of invasive phragmites (common reed) in gulf of mexico coastal wetlands using multispectral imagery and small unmanned aerial systems. *International Journal of Remote Sensing*, Informa UK Limited, p. 1–22, dec 2016.

SCHWENKER, F.; KESTLER, H. A.; PALM, G. Three learning phases for radial-basis-function networks. *Neural Networks*, Elsevier BV, v. 14, n. 4-5, p. 439–458, may 2001. <https://doi.org/10.1016/s0893-6080(01)00027-2>.

SCOTT, S. M.; JAMES, D.; ALI, Z. Data analysis for electronic nose systems. *Microchimica Acta*, Springer Nature, v. 156, n. 3-4, p. 183–207, sep 2006.

SEDHURAMAN, K.; HIMAVATHI, S.; MUTHURAMALINGAM, A. Comparison of learning algorithms for neural network based speed estimator in sensorless induction motor drives. In: *IEEE-International Conference On Advances In Engineering, Science And Management (ICAESM -2012)*. [S.l.: s.n.], 2012. p. 196–202.

SHAFRI, H. Z. M.; SUHAILI, A.; MANSOR, S. The performance of maximum likelihood, spectral angle mapper, neural network and decision tree classifiers in hyperspectral image analysis. *Journal of Computer Science*, Science Publications, v. 3, n. 6, p. 419–423, jun 2007.

SHEN, L.; BAI, L.; FAIRHURST, M. Gabor wavelets and general discriminant analysis for face identification and verification. *Image and Vision Computing*, Elsevier BV, v. 25, n. 5, p. 553–563, may 2007.

SIMONITE, T. *Moore's Law Is Dead. Now What?* 2016. <https://www.technologyreview.com/s/601441/moores-law-is-dead-now-what/>.

SINGLA, R. et al. Comparison of SVM and ANN for classification of eye events in EEG. *Journal of Biomedical Science and Engineering*, Scientific Research Publishing, Inc„ v. 04, n. 01, p. 62–69, 2011.

SOH, L. K.; TSATSOULIS, C. Texture analysis of sar sea ice imagery using gray level co-occurrence matrices. *IEEE Transactions on*

Geoscience and Remote Sensing, v. 37, n. 2, p. 780–795, Mar 1999. ISSN 0196-2892.

SOLA, J.; SEVILLA, J. Importance of input data normalization for the application of neural networks to complex industrial problems. *IEEE Transactions on Nuclear Science*, v. 44, n. 3, p. 1464–1468, Jun 1997. ISSN 0018-9499.

SPECHT, D. F. Probabilistic neural networks. *Neural Netw.*, Elsevier Science Ltd., Oxford, UK, UK, v. 3, n. 1, p. 109–118, jan. 1990. ISSN 0893-6080.

SRINIVASARAO, B.; SREENIVASAN, G.; SHARMA, S. Compensation of voltage distribunces in smib system using ann based dpfc controller. In: *2016 International Conference on Signal Processing, Communication, Power and Embedded System (SCOPES)*. [S.l.: s.n.], 2016. p. 563–567.

STANKOVIĆ, R. S.; FALKOWSKI, B. J. The haar wavelet transform: its status and achievements. *Computers & Electrical Engineering*, Elsevier BV, v. 29, n. 1, p. 25–44, jan 2003.

TEAM, E. G. C. P. C. *East Gulf Coastal Plain Ecoregional Plan. Volume I and II (Revision)*. Arlington, Virginia, 2001. 36 p.

THAKUR, A.; MISHRA, D. Hyper spectral image classification using multilayer perceptron neural network functional link ann. In: *2017 7th International Conference on Cloud Computing, Data Science Engineering - Confluence*. [S.l.: s.n.], 2017. p. 639–642.

The Pennsylvania State University. *Neurons*. 2017. <*https* : *//online.science.psu.edu/bisc*004_*activewd*001/*node*/1907>.

TU, J. V. Advantages and disadvantages of using artificial neural networks versus logistic regression for predicting medical outcomes. *Journal of Clinical Epidemiology*, Elsevier BV, v. 49, n. 11, p. 1225–1231, nov 1996.

VIERA, A. J.; GARRETT, J. M. Understanding interobserver agreement: The kappa statisti. *Journal of Family Medicine*, p. 360–363, may 2005.

VILLA, P. et al. A remote sensing approach to monitor the conservation status of lacustrine phragmites australis beds. *Wetlands Ecology and Management*, Springer Nature, v. 21, n. 6, p. 399–416, jul 2013.

WANG, X.; SHI, B. E. GPU implemention of fast gabor filters. In: *Proceedings of 2010 IEEE International Symposium on Circuits and Systems*. IEEE, 2010. <https://doi.org/10.1109/iscas.2010.5537757>.

XIE, T.; YU, H.; WILAMOWSKI, B. Comparison between traditional neural networks and radial basis function networks. In: *2011 IEEE International Symposium on Industrial Electronics*. [S.l.: s.n.], 2011. p. 1194–1199. ISSN 2163-5137.

YU, H.; WILAMOWSKI, B. M. Intelligent systems (industrial electronics). In: _____. [S.l.]: CRC Press, 2011. cap. Levenberg–Marquardt Training.

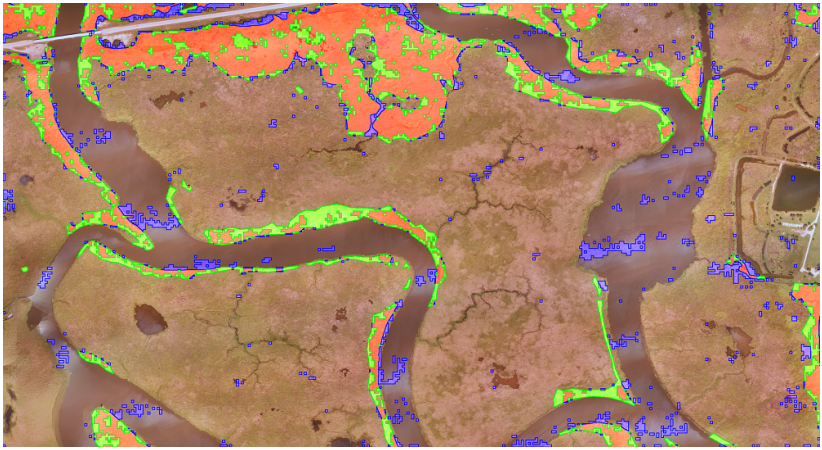
YUAN, H.; WIELE, C. F. V. D.; KHORRAM, S. An automated artificial neural network system for land use/land cover classification from landsat TM imagery. *Remote Sensing*, MDPI AG, v. 1, n. 3, p. 243–265, jul 2009.

ZAMAN, B.; JENSEN, A. M.; MCKEE, M. Use of high-resolution multispectral imagery acquired with an autonomous unmanned aerial vehicle to quantify the spread of an invasive wetlands species. In: *2011 IEEE International Geoscience and Remote Sensing Symposium*. [S.l.: s.n.], 2011. p. 803–806. ISSN 2153-6996.

ZHAO, W.-B.; HUANG, D.-S.; GUO, L. Comparative study between radial basis probabilistic neural networks and radial basis function neural networks. In: *Intelligent Data Engineering and Automated Learning*. [S.l.]: Springer Berlin Heidelberg, 2003. p. 389–396.

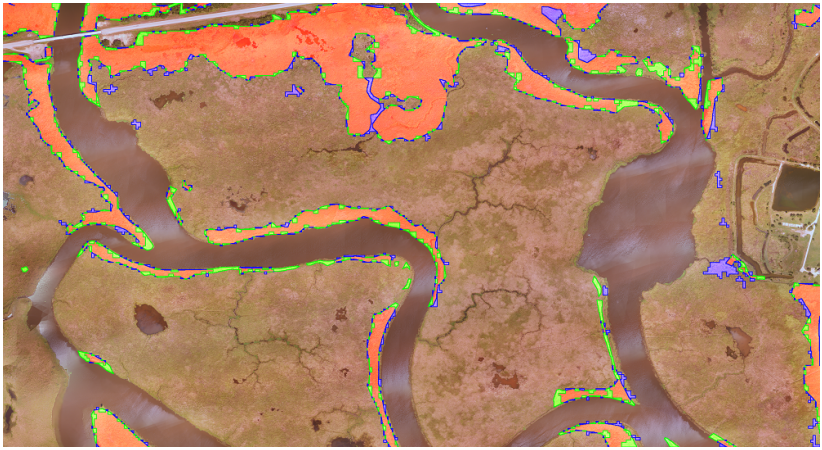
ANNEX A – Gabor Extracted Features Examles

Figure 42 – Site I assessment for MLP classification using texture features based on GF.



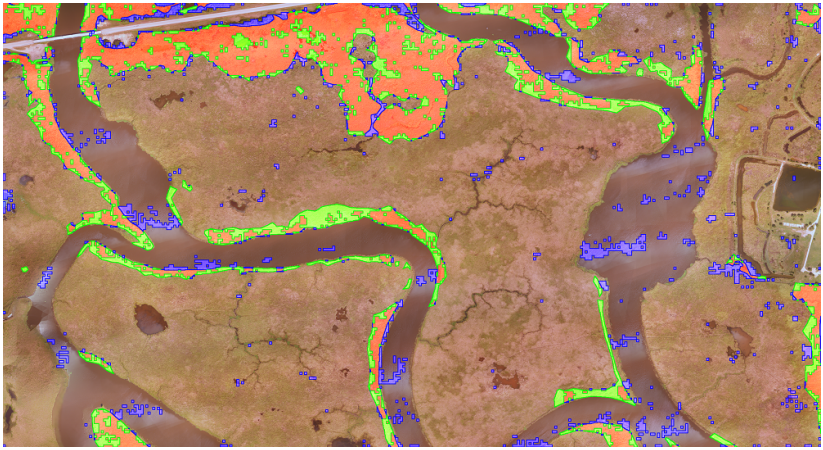
Source: Image from authors.

Figure 43 – Site II assessment for PNN classification using SFTA texture features.



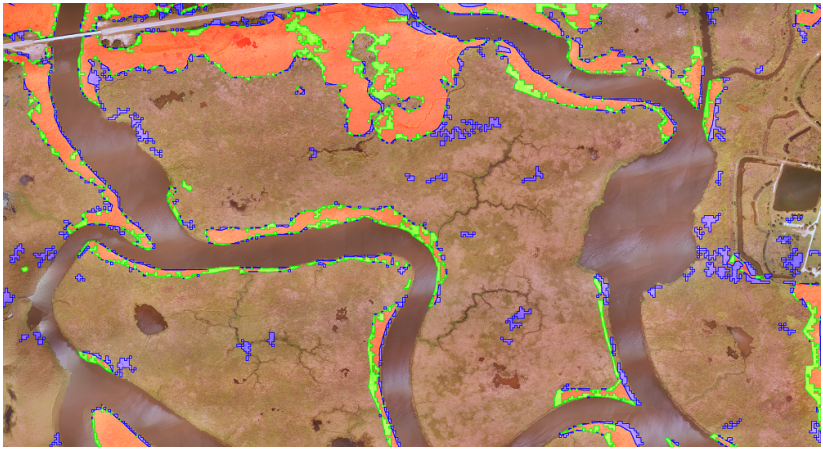
Source: Image from authors.

Figure 44 – Site I assessment for MLP classification using SFTA texture features.



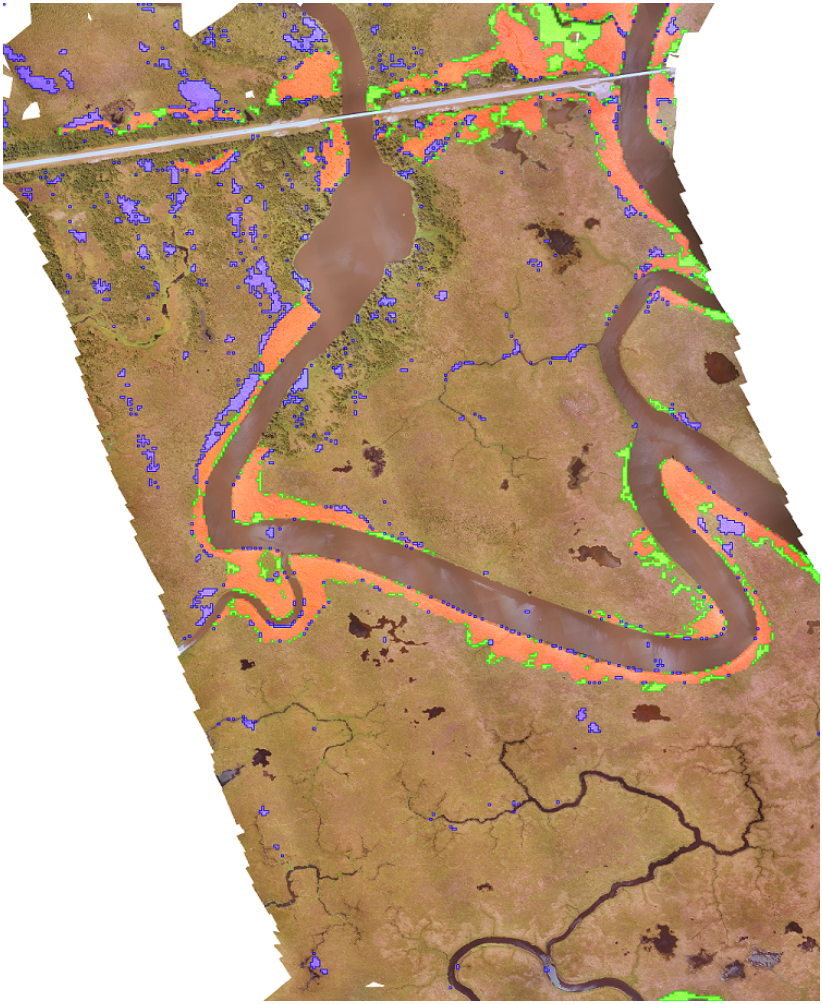
Source: Image from authors.

Figure 45 – Site I assessment for RBFN classification using WTA.



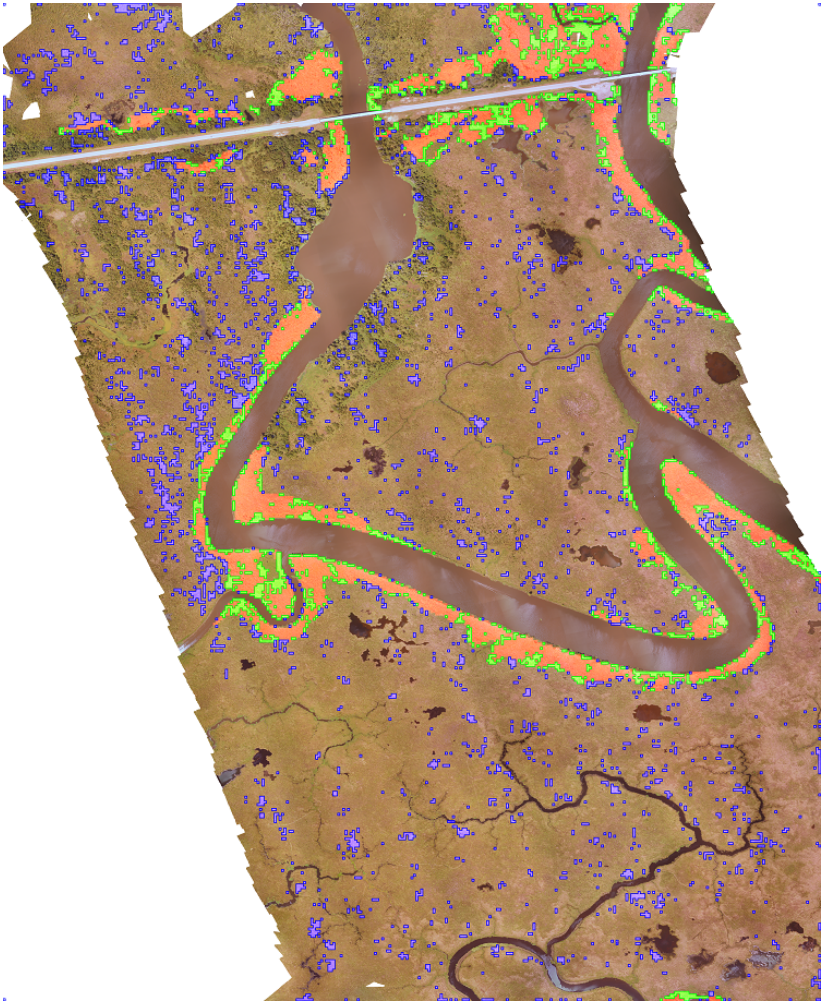
Source: Image from authors.

Figure 46 – Site II assessment for MLP classification using GLCM.



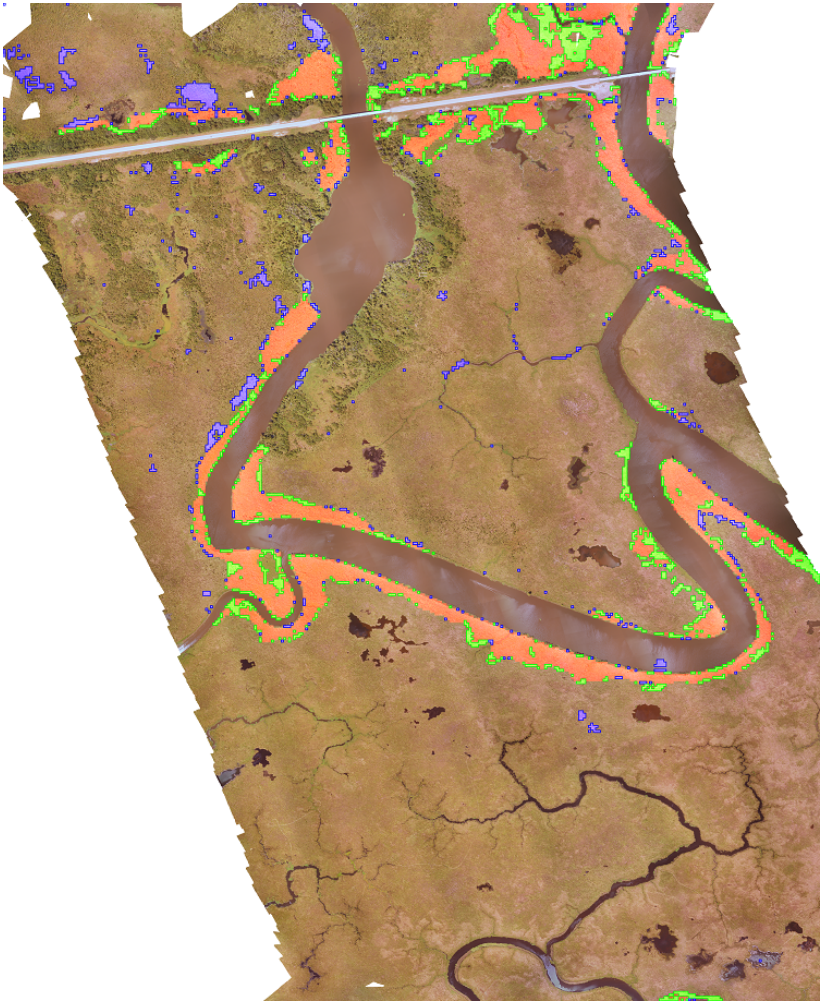
Source: Image from authors.

Figure 47 – Site II assessment for MLP classification using GF.



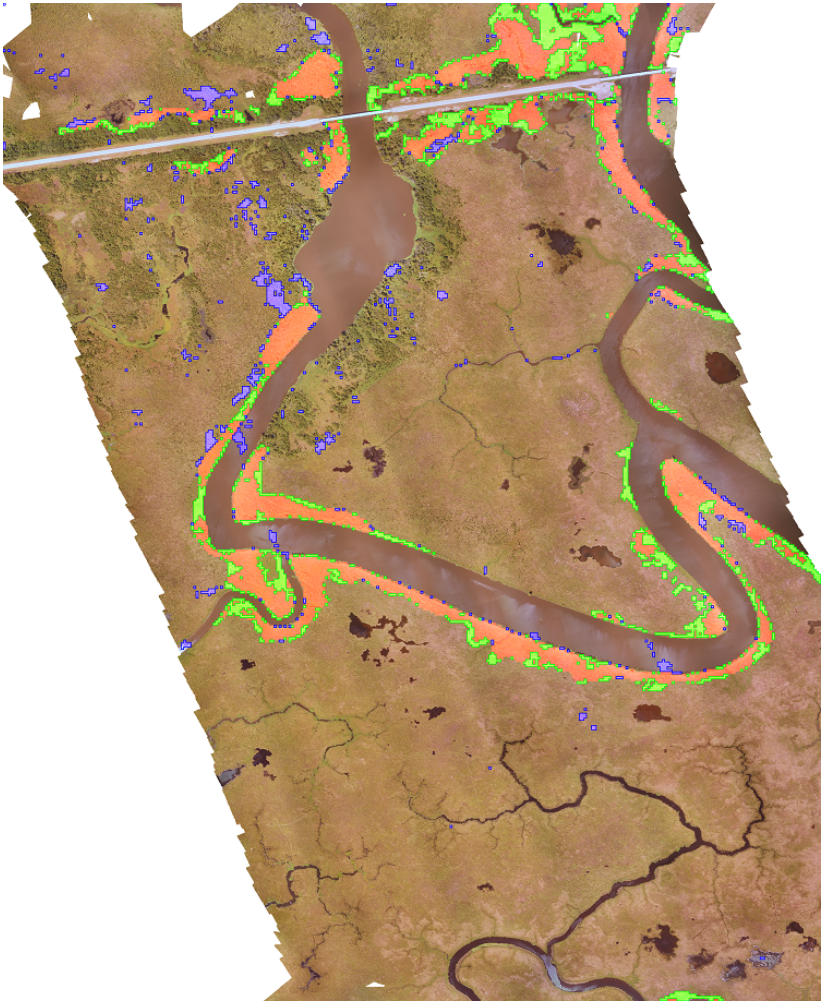
Source: Image from authors.

Figure 48 – Site II assessment for MLP classification using SFTA.



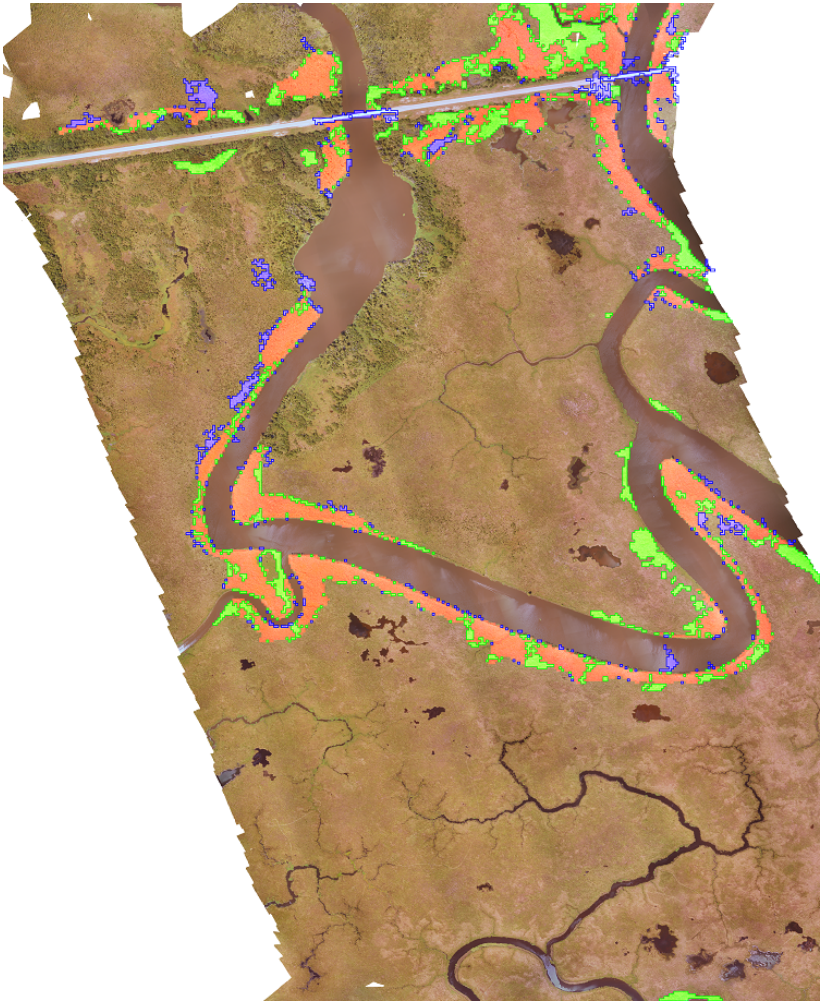
Source: Image from authors.

Figure 49 – Site II assessment for MLP classification using WTA.



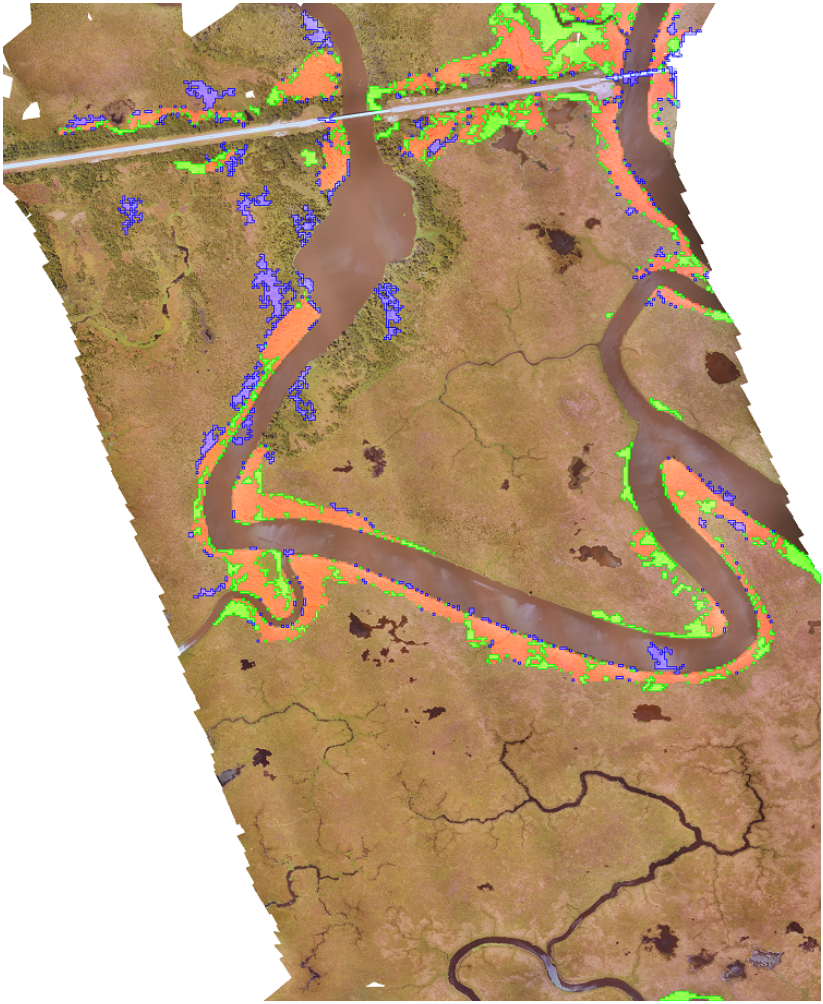
Source: Image from authors.

Figure 50 – Site II assessment for RBFN classification using GLCM.



Source: Image from authors.

Figure 51 – Site II assessment for RBFN classification using WTA.



Source: Image from authors.