UNIVERSIDADE DE BRASÍLIA

INSTITUTO DE GEOCIÊNCIAS – IG
PÓS-GRADUAÇÃO EM GEOCIÊNCIAS APLICADAS E GEODINÂMICA

INTEGRAÇÃO, CONTROLE E ACOMPANHAMENTO DA ANÁLISE DE IMAGENS BASEADA EM OBJETO E MINERAÇÃO DE DADOS POR MEIO DA PLATAFORMA DISTRIBUÍDA INTERCLOUD

Área de Concentração: Geoprocessamento e Análise Ambiental

TESE DE DOUTORADO N° 39

RODRIGO RODRIGUES ANTUNES

BRASÍLIA – DF

ABRIL/2018

UNIVERSIDADE DE BRASÍLIA

INSTITUTO DE GEOCIÊNCIAS – IG
PÓS-GRADUAÇÃO EM GEOCIÊNCIAS APLICADAS E GEODINÂMICA

INTEGRAÇÃO, CONTROLE E ACOMPANHAMENTO DA ANÁLISE DE IMAGENS BASEADA EM OBJETO E MINERAÇÃO DE DADOS POR MEIO DA PLATAFORMA DISTRIBUÍDA INTERCLOUD

Rodrigo Rodrigues Antunes

Orientador: Professor Dr. Edilson de Souza Bias

Co-orientador do Brasil: Prof. Dr. Gilson Alexandre Ostwald Pedro da Costa
(Pontifícia Universidade Católica do Rio de Janeiro)

Co-orientador (Sanduíche): Prof. Dr. Thomas Blaschke
(Universidade de Salzburg)

Tese apresentada a Banca Examinadora do Curso de Doutorado do Programa de Pós-Graduação da Universidade de Brasília (UnB), como requisito parcial para obtenção do grau de **Doutor em Geociências** na Área de Concentração de Geoprocessamento e Análise Ambiental.

BRASÍLIA – DF

ABRIL/2018

UNIVERSIDADE DE BRASÍLIA

INSTITUTO DE GEOCIÊNCIAS – IG
PÓS-GRADUAÇÃO EM GEOCIÊNCIAS APLICADAS E GEODINÂMICA

INTEGRAÇÃO, CONTROLE E ACOMPANHAMENTO DA ANÁLISE DE IMAGENS BASEADA EM OBJETO E MINERAÇÃO DE DADOS POR MEIO DA PLATAFORMA DISTRIBUÍDA INTERCLOUD

Rodrigo Rodrigues Antunes

BANCA EXAMINADORA

Prof. Dr. Edilson de Souza Bias - IGD/UnB
Presidente

Prof. Dr. Gustavo Baptista Macedo de Melo - IGD/UnB
Membro Interno

Prof. Dr. Raul Queiroz Feitosa – PUC-Rio
Membro Externo

Prof. Dr. José Alberto Quintanilha - USP
Membro Externo

BRASÍLIA – DF

ABRIL/2018

Ao meu filho Rafael, que é o principal motivador e por suportar minha ausência por dedicação a esta pesquisa.

AGRADECIMENTOS

Agradeço a todos que me auxiliaram na execução desta tese e em especial: ao professor Edilson Bias pela orientação, parceria, confiança, amizade, paciência e dedicação. Com todas essas características, conseguimos atravessar fronteiras do conhecimento juntos, principalmente com bom trabalho desenvolvido na Universidade de Salzburg, Áustria.

Ao co-orientador Prof. Dr. Gilson Alexandre Ostwald Pedro da Costa, da Pontifícia Universidade Católica do Rio de Janeiro. Foi um dos principais incentivadores desta pesquisa.

A todos os meus professores e colaboradores do Instituto de Geociências da Universidade de Brasília, que, de alguma forma, contribuíram para alcançar o objetivo final, a tese.

A CAPES, pelo auxílio financeiro prestado para Doutorado Sanduíche na Universidade de Salzburg, Áustria.

Aos professores Dr. Thomas Blaschke e Dr. Dirk Tiede, da Universidade de Salzburg, Áustria, que a mim concederam boa estada naquele país e contribuíram com seus conhecimentos no desenvolvimento deste trabalho.

A minha esposa Karyne, família e amigos por acreditarem no desafio a mim concedido. Em especial a minha mãe, Joalcema, pela educação a mim concedida para me tornar quem eu sou.

À memória do meu pai, Claudionor, por incentivar o conhecimento desde o início de minha vida.

RESUMO

Atualmente, enormes volumes de dados de sensoriamento remoto são geradas em pouco espaço de tempo e manipular esses dados se torna um desafio para os profissionais e pesquisadores de sensoriamento remoto (SR), que necessitam de ferramentas e modelos mais eficientes de processamento e interpretação de imagens. Nesta linha de raciocínio, o presente trabalho apresenta um novo método *on-line* de integração de uma plataforma distribuída de classificação de imagem baseada em objetos e algoritmo de classificação de aprendizado de máquina para criação de modelos estatísticos de interpretação. Por meio do sistema InterCloud, que é uma nova plataforma de interpretação de imagens projetada para rodar em redes de computadores (clusters físicos ou infra-estrutura de computação em nuvem), e os frameworks para computação distribuída Apache Hive que cria tabelas virtuais, a MLlib do Apache Spark que é uma biblioteca de machine learning e o Apache Zeppelin que disponibiliza um notebook web, foi possível disponibilizar dados, tabelas e gráficos com valores de pixels para modelagem estatísticas de interpretação. No protótipo implementado, o sistema Apache Zeppelin forneceu os meios para usar a biblioteca de aprendizado de máquina Scikit-Learn Python na criação de um modelo de classificação (Árvore de Decisão), que foi simulado no InterCloud por meio de um sript pig. Neste trabalho, também avaliamos a abordagem com uma aplicação de interpretação de imagem baseada em objeto, cobertura terrestre, realizada em uma cena GeoEye-1 de 103 Km² (19k por 23k pixels), usando recursos de um serviço de infraestrutura de computação em nuvem comercial. 24 atributos (espectrais e morfológicos) e 11 classes de objetos, incluindo alvos urbanos e rurais, foram considerados. O estudo avaliou as possibilidades de escalabidade para execução de diferentes tarefas e, a exatidão da classificação por meio de uma matriz de confusão.

Palavras-chaves: Análise de Imagem Baseada em Objetos, Computação em Nuvem, Mineração de Dados, InterCloud, Apache Zeppelin.

ABSTRACT

Currently, huge amounts of remote sensing data are generated in a short time and manipulating such data becomes a challenge for Remote Sensing (SR) professionals and researchers. Efficient tools and patterns of image processing and interpretation need to be made available.

The present study is aimed to show a new online method of integrating a distributed object-based image classification platform and machine learning Decision Tree algorithm for creating statistical patterns of interpretation.

Through the InterCloud system, which is a new imaging platform designed to run on computer networks (physical clusters or cloud computing support), and the Apache Hive distributed computing frameworks that create virtual tables, MLlib of Apache Spark which is a library of machine learning and Apache Zeppelin that makes available a web notebook, it was possible to make available data, tables, and graphics with pixel values for statistical patterns of interpretation.

In the prototype implemented, the Apache Zeppelin system provided the means to use another Sci-kit-Learn Python machine learning library establishing a classification pattern (Decision Tree) that was simulated in InterCloud platform by means of a script pig.

We also used the object-based image analysis approach interpretation to evaluate the image into terrestrial coverage, performed in a 103 Km² (19k by 23k pixels) GeoEye-1 scene using features of a commercial cloud computing support service.

24 attributes (spectral and morphological) and 11 classes of objects, including urban and rural targets, were considered.

In addition to the accuracy of the classification result evaluated by means of accurate indexes, we evaluate the InterCloud ability to perform different tasks (distributed segmentation, extraction of characteristics and distributed classification) with different configurations of the cloud infrastructure, in which they were varied in the number of nodes/clusters.

The accuracy index of the final classification was evaluated by means of the confusion matrix in agreement with the coefficients.

Keywords: Object-Based Image Analysis, Cloud Computing, Data Mining, InterCloud, Apache Zeppelin.

LISTA DE SIGLAS E ABREVIATURAS

AWS	Amazon Web Services
CHAID	Chi-square Automatic Interaction Detector
CART	Classification and Regression Trees for Machine Learning
GEOBIA	Geographic Object Based Image Analysis
HDFS	Hadoop Distributed File System
ID3	Iterativo DiChaudomiser 3
JSON	JavaScript Object Notation
KDD	Knowledge Discovery in Databases
MLlib	Machine learning library
NDVI	Normalized Difference Vegetation Index
OBIA	Object Based Image Analysis
QGIS	Open Source Geographic Information System
SR	Sensoriamento Remoto
SQL	Structured Query Language
UDF	User Defined Functions
WEKA	Waikato Environment for Knowledge Analysis

“Não desejaria, com minha obra, poupar aos outros o trabalho de pensar,
mas sim, se for possível, estimular alguém a pensar por si próprio.”
(WITTGENSTEIN, 1975, prefácio).

LISTA DE ILUSTRAÇÕES

LISTA DE TABELAS

SUMÁRIO

1 - INTRODUÇÃO

O impacto no uso do solo concernente à dinâmica populacional no mundo procede em todas as áreas habitadas e de várias formas (RUFINO; SILVA, 2017). Atualmente, o avanço das ocupações intra-urbanas em cidades de pequeno, médio e grande porte representa grande desafio para os governantes. As cidades, por meio de seus administradores públicos, precisam de dados reais e ferramentas automatizadas para acompanhar e fiscalizar a expansão e a alteração intra-urbana e, assim, obter respostas rápidas e consistentes para o desenvolvimento de planejamento urbano eficiente. Lang (2008) corrobora com essa visão, descrevendo que há demanda cada vez maior para a regularidade de atualização das informações geo-espaciais, combinada com técnicas de extração rápida, para melhor processo de tomada de decisão e, consequentemente, de gestão.

Atualmente, com o avanço dos sistemas de sensores de alta resolução espacial e espectral, assumindo relevância na representação de informações sobre a superfície terrestre (LIU, 2015), vinculado à utilização da Análise de Imagem Baseada em Objeto (OBIA), reconhecida por Blaschke et al. (2014) como novo paradigma em sensoriamento remoto, é possível identificação mais precisa dos alvos urbanos e intra-urbanos. A diversidade e a realidade dos objetos inferidas por meio desses sensores podem ser mais bem interpretadas com o uso de OBIA, devido à possibilidade de introduzir o conhecimento do analista e de atribuir vários parâmetros além da já consagrada resposta espectral.

Segundo Blaschke (2010), por volta do ano 2000, o SIG (Sistemas de Informação Geográfica) e o processamento de imagem começaram a crescer rapidamente por meio da OBIA ou GEOBIA (Geographic Object-Based Image Analysis).

A OBIA veio superar problemas das técnicas tradicionais baseadas unicamente na análise pixel-a-pixel das imagens de alta resolução espacial, definindo, em primeiro lugar, os segmentos, em vez de pixels, permitindo variabilidade de atributos além da reflectância espectral para discriminar características de objetos (BLASCHKE; TOMLJENOVIĆ, 2012).

Algumas utilizações com base em OBIA para o planejamento urbano já foram desenvolvidas em diversos locais do planeta, apresentando resultados bastante satisfatórios.

Chen e Chen (2014), reconhecendo a limitação dos algoritmos tradicionais com base em pixel para classificação de área urbana, trabalharam na detecção de mudança baseada em objeto com dados do WorldView-2 para monitoramento urbano em Pequim, China. Os resultados da pesquisa indicaram que o método baseado em objeto melhorou significativamente a precisão da detecção de mudança, e que 12% da área, incluindo edifícios, foram alterados (2012-2013). A acurácia global chegou próximo de 89% e o coeficiente Kappa de 65%.

Orlando e La Rosa (2014) apresentaram metodologia de classificação orientada a objetos para detectar e analisar dados de satélites multi-temporais (2002 e 2006) de Scopello-Sicily, Itália. Foi utilizado o *software* eCognition para classificação e chegou-se à acurácia de 94,30% (cobertura de cimento).

Bias et al. (2014) utilizaram a OBIA para monitoramento do cadastro urbano na cidade de Goianésia, no Brasil. Os resultados foram avaliados por meio do índice de exatidão TAU, com concordância de 84%. O *software* utilizado para a classificação foi o InterIMAGE, que permitiu a mineração de dados para a construção da árvore de decisão por meio do minerador WEKA.

Uma ferramenta que vem destacando-se no contexto de classificação de imagens de alta resolução espacial é o InterIMAGE, plataforma livre, baseada em conhecimento para interpretação automática de imagens (InterIMAGE, 2010), que possui alguns operadores baseado em OBIA no processo de classificação. Porém, a versão atual possui limitação de processamento para grandes imagens (até 3.000 x 3.000 pixels).

A era atual é de *big data,* e *petabytes* de dados são gerados diariamente (TSAI et al., 2016; LIU, 2015). O projeto como EOSDIS da NASA, por exemplo, produz cerca de 12 TB de dados diariamente (NASA, 2017). Este cenário leva a novos desafios relacionados à capacidade de lidar com enormes volumes de dados em relação a técnicas e recursos computacionais (LEE; KANG, 2015). E, nesse sentido, o tratamento de dados de sensoriamento remoto pode ser considerado problema, devido ao grande volume desses dados, variedade e velocidade de

geração (MA et al., 2015; JADHAV, 2013). Com a necessidade de classificar grande volume de dados para obtenção de visão espacial para as diversas aplicações necessárias à tomada de decisões, busca-se desenvolver mecanismos que expandam a utilização de OBIA em ambiente de computação distribuída.

Uma nova plataforma, denominada InterCloud (InterIMAGE Cloud Platform), está sendo desenvolvida e testada e resultará na reformulação do sistema atual (COSTA et al., 2010). O InterCloud é uma plataforma livre para interpretação de imagens, projetada para executar *grids* de computadores (*cluster* físico ou infraestrutura de computação em nuvem) e permitir a interpretação de grandes conjuntos de dados de sensoriamento remoto de forma eficiente (FERREIRA, 2015). Esse sistema também possui alguns operadores para aplicar OBIA na classificação distribuída.

Importante observar a limitação dos sistemas InterIMAGE e InterCloud para classificação totalmente baseada em OBIA. As vantagens típicas que caracterizam OBIA como contexto, vizinhança topológica e distâncias ainda não foram implementadas em seu conjunto de operadores.

Em função da grande variedade de atributos disponíveis, principalmente em ambientes urbanos, os modelos de classificação baseados em objetos podem tornar-se muito complexos para serem definidos unicamente a partir de conhecimento empírico. Nesse contexto, técnicas de mineração de dados podem auxiliar, tornando possível explorar o grande potencial discriminatório dos atributos e adquirir conhecimento acerca da relação entre as variáveis e as classes de objetos de interesse.

A integração de sistema de mineração de dados com vários sistemas de classificação, entre eles InterIMAGE e eCognition, tem apresentado bons resultados de acordo com os estudos desenvolvidos, por exemplo:

I. Uso de algoritmos de classificação J48 para mapear o uso e a cobertura do solo (FARIA et al., 2014);
II. Aplicação de técnicas de mineração de dados e GEOBIA para análise de susceptibilidade ao fogo no Parque Nacional Itatiaia, no Brasil (SOUSA; FERNANDES; COSTA, 2014);

III. Análise do nível de detalhe nas classificações de áreas urbanas com VHR óptico e imagens hiperespectrais usando método não paramétrico (ANJO; ALMEIDA; GALVÃO, 2014);
IV. Abordagem de árvore de decisão para classificação de imagem de sensoriamento remoto (SHARMA; GHOSH; JOSHI, 2013);
V. Classificação orientada a objeto em associação às ferramentas reflectância acumulada e mineração de dados a objeto em associação às ferramentas reflectância acumulada e mineração de dados (DE GRANDE et al., 2017);
VI. Desenvolvimento de técnica para monitoramento do cadastro urbano baseado em classificação orientada a objetos. Estudo de caso: município de Goianésia – Goiás (ANTUNES et al., 2014);
VII. Classificação da cobertura da terra utilizando os programas livres: InterIMAGE, WEKA e QuantumGIS (NASCIMENTO et al., 2013);
VIII. Classificação de cobertura de terra da planície de inundação do lago Grande de Curuai (Amazonas, Brasil), usando técnicas de fusão multi-sensor e imagem (FURTADO et al., 2015);
IX. Integração de ferramentas de código aberto para o monitoramento baseado em objetos de alvos urbanos (ANTUNES et al., 2016);
X. Análise do nível de legenda de classificação de áreas urbanas empregando imagens multiespectrais e hiperespectrais com os métodos árvore de decisão C4.5 e floresta randômica (DOS ANJOS et al., 2017).

Plataformas *open source* para processos de classificação baseado em OBIA estão sendo muito utilizadas e vem se destacando a cada dia, principalmente em países onde as pesquisas e universidades encontram dificuldades e limitações financeiras, como no próprio Brasil.

Até o presente momento, somente o sistema de mineração de dados WEKA (*Waikato Environment for Knowledge Analysis*) é utilizado pelos analistas para integração com o InterIMAGE. Testes atuais com outros sistemas mineradores, Orange Canvas e SIPINA, estão sendo abordados nesta pesquisa e também mostraram bons resultados até o momento (ANTUNES et al., 2016; ANTUNES et al., 2018).

Atualmente, os métodos de classificação supervisionada no InterCloud são implementados utilizando-se somente funções do WEKA para classificação e baseiam-se unicamente na execução de algoritmos de aprendizagem de máquina. Não há modelo de interpretação ou conhecimento, nem visualização gráfica dos dados.

Contudo, a presente tese enfoca novo método de integração, controle e acompanhamento de sistema distribuído de classificação de imagem orbital baseado em OBIA com bibliotecas de aprendizado de máquina na nuvem.

Para a implementação do novo método foram utilizadas plataforma distribuída de interpretação de imagem InterCloud (*open source*) e biblioteca de aprendizagem de máquina MLlib Spark e *Python Learning*. O método de classificação utilizado foi de árvore de decisão. O *framework* Apache Hive permitiu criar tabelas virtuais em *cluster* e o Apache Zeppelin, que é um *notebook web* para interpretação de códigos em diversas linguagens (Scala, Python e outros) em ambiente de computação distribuída, permitiu visualizar e modelar os dados com valores de pixels por meio de gráficos e consultas SQL.

OBJETIVOS

Os objetivos principais da presente tese é propor e desenvolver modelo para análise de dados e definição de limiares de classificação (OBIA), em cluster, visando integrar plataforma distribuída de interpretação de imagem orbital.

Os objetivos específicos deste trabalho são:

a) identificar formas de integrar, em *cluster*, o sistema distribuído de interpretação de imagem orbital e bibliotecas de aprendizagem de máquina, visando expor graficamente os dados de classificação;

b) analisar o desempenho computacional e o resultado da classificação de árvore de decisão por meio da interface gráfica disponibilizada pelo modelo proposto;

c) analisar procedimentos para subsidiar a classificação de grande imagem orbital de alta resolução espacial com base no resultado da classificação de árvore de decisão; e

d) propor procedimentos de qualidade com base na aplicação dos índices de exatidão Global, Kappa e TAU.

PROBLEMA

A falta de interação entre o operador e os dados classificados e processados em nuvem geram dificuldades e limitações na melhoria e no controle dos processos e, consequentemente, nos resultados.

HIPÓTESE

A implementação dos processos de aprendizado de máquina executados em cluster e a integração com sistema de classificação de imagens distribuídas (InterCloud) permitirão melhor qualidade e agilidade nas tarefas de construção de árvores e regras de decisão em ambiente de processamento totalmente distribuído, permitindo a classificação (OBIA) de grande imagem orbital de forma mais efetiva.

CONTRIBUIÇÕES ORIGINAIS DA TESE

Esta tese apresenta novo método de integração, em *cluster*, de sistema distribuído de classificação de imagem e aprendizagem de máquina para classificação baseada em objetos (OBIA). As principais contribuições são:

a) primeira aplicação completa, baseada em OBIA, utilizando grande imagem de alta resolução com sistema distribuído de interpretação de imagem (Intercloud);
b) avaliação de outros *frameworks*, além do Hadoop e Pig, para integrar ao sistema distribuído de classificação de imagem InterCloud. Os *frameworks* testados e avaliados foram: Apache Hive, Apache Spark e Apache Zeppelin;
c) avaliação de outras bibliotecas de aprendizagem de máquina, além do WEKA, para integrar ao sistema distribuído de classificação (InterCloud). As outras bibliotecas utilizadas foram: MLlib Spark e Python Learning.

d) novo método de analisar e qualificar dados, tabelas e gráficos com valores de pixel por meio de *interface web*, executado em *cluster*, possibilitando criar modelos estatísticos, tais como: gráficos de dispersão, definição de limites, árvores de decisão e outros.

ESTRUTURA DA TESE

Esta pesquisa encontra-se dividida em 8 seções, organizada da seguinte forma:

Seções 1.1, 1.2, 1.3, 1.4 e 1.5 – fazem parte da introdução da tese e apresentam os objetivos, hipótese e a problemática da pesquisa.

Faz parte da estrutura da tese três artigos submetidos e/ou publicados em revistas nacionais e internacionais. Os artigos encontram-se nas seções 2,3 e 4:

Seção 2 – ARTIGO 1

Título: **OBJECT-BASED ANALYSIS FOR URBAN LAND COVER MAPPING USING THE INTERIMAGE AND THE SIPINA FREE SOFTWARE PACKAGES**
Artigo publicado em: Boletim de Ciências Geodésicas (Qualis – B1)
Data da publicação: janeiro de 2018
Disponível em: https://revistas.ufpr.br/bcg/article/view/58630 <acesso: 04/05/2015>
Comprovante da submissão: Anexo 1

Seção 3 – ARTIGO 2

Título: **ANÁLISE DE SISTEMAS MINERADORES DE DADOS OPEN-SOURCE E SEUS ALGORITMOS DE CLASSIFICAÇÃO DE ÁRVORE DE DECISÃO INTEGRADOS COM O SISTEMA DE CLASSIFICAÇÃO BASEADA EM OBJETOS INTERIMAGE**
Artigo submetido em: Revista Brasileira de Cartografia (Qualis – B2)
Data da submissão: novembro de 2017
Comprovante da submissão: Anexo 2

Seção 4 – ARTIGO 3

Título: **PROOF OF CONCEPT OF A NOVEL CLOUD COMPUTING APPROACH FOR OBJECT-BASED REMOTE SENSING DATA ANALYSIS AND CLASSIFICATION**

Artigo submetido em: GIScience Remote Sensing (Fator de Impacto = 3,049)

Data da submissão: fevereiro de 2018

Comprovante da submissão: anexo 3

Seção 5 – são discutidos os resultados referentes à pesquisa.

Seção 6 – são apresentadas a conclusão e recomendações para futuras pesquisas.

Seção 7 – são apresentadas as referências bibliográficas gerais utilizadas na tese, levando em consideração as referências dos artigos 1, 2 e 3.

Seção 8 – estão anexados os comprovantes das submissões dos artigos 1,2 e 3, desta tese.

2 - ARTIGO 1

OBJECT-BASED ANALYSIS FOR URBAN LAND COVER MAPPING USING THE INTERIMAGE AND THE SIPINA FREE SOFTWARE PACKAGES

Análise baseada em objeto para mapeamento do uso do solo urbano utilizando os pacotes de software InterIMAGE e SIPINA

Rodrigo Rodrigues Antunes[1]
Edilson de Sousa Bias[1]
Gilson Alexandre Ostwald Pedro da Costa[2]
Ricardo Seixas Brites[1]

[1] Instituto de Geociências, Universidade de Brasília.
Campus Universitário Darcy Ribeiro, Brasília, Distrito Federal, Brasil, 70910-900
rodrigorantunes@hotmail.com, edbias@unb.br

[2] Instituto de Matemática e Estatística, Universidade Estadual do Rio de Janeiro.
R. São Francisco Xavier, 524 - Maracanã, Rio de Janeiro - RJ, Brasil, 20550-900
rodrigorantunes@hotmail.com, gilson.costa@ime.uerj.br

Abstract:

In this work we introduce an object-based method, applied to urban land cover mapping. The method is implemented with two open-source tools: SIPINA, a data mining software package; and InterIMAGE, an object-based image analysis system. Initially, segmentation, feature extraction and sample selection procedures are performed with InterIMAGE. In order to reduce the time and subjectivity involved to develop the decision rules in InterIMAGE, a data mining step is then carried out with SIPINA. In sequence, the decision trees delivered by SIPINA are analysed and encoded into InterIMAGE decision rules for the final classification step. Experiments were conducted using a subset of a GeoEye image, acquired in January 01, 2013, covering the urban portion of the municipality of Goianésia, Brazil. Five decision tree induction algorithms, available in SIPINA, were tested: ID3, C45, GID3, Assistant86 and CHAID. The TAU and Kappa coefficients were used to evaluate the results. The TAU values obtained were in the range of 0.66 and 0.70, while those for Kappa varied from 0.65 to 0.69.

Keywords: Object-Based Image Analysis, Data Mining, InterIMAGE, SIPINA

Resumo:

Apresentamos neste trabalho um método para o mapeamento do uso do solo urbano, implementado com duas ferramentas de código aberto: SIPINA, um pacote de *software* de mineração de dados; e o InterIMAGE, um sistema de análise de imagens de sensoriamento remoto baseado em objetos. Inicialmente procedimentos de segmentação, extração de atributos e seleção de amostras são realizados com o InterIMAGE. Com o objetivo de reduzir o tempo e a subjetividade envolvidos na definição de regras de decisão no InterIMAGE, um procedimento de mineração de dados é então realizado com a SIPINA. Na sequência, as árvores de decisão geradas através do SIPINA são analisadas e codificadas em regras de decisão do InterIMAGE para o procedimento final de classificação. Experimentos foram realizados sobre uma imagem GeoEye, recobrindo uma paisagem urbana do município de Goianésia, Brasil. Foram testados cinco algoritmos de indução de árvores de decisão disponíveis no SIPINA: ID3, C45, GID3, Assistant86 e CHAID. Os resultados foram avaliados através dos índices TAU e Kappa. Os valores de TAU obtidos variaram entre 0.66 e 0.70, e os valores de Kappa variaram entre 0.65 e 0.69.

INTRODUCTION

The impact of land use on the population dynamics occurs in all inhabited areas in many different ways (Rufino and Silva, 2017). Small, medium or large size cities need up-to-date data and automated tools to monitor and regulate urban expansion in order to ensure quick and consistent solutions towards efficient urban planning.

According to Cerqueira and Alves (2010), the number of remote sensing applications for urban environments have increased over the last decade, resulting in advances in large scale mapping, which is an extremely useful tool for urban planning and to manage the unregulated growth of urban areas, noticeably in developing countries.

Advanced high spatial and spectral resolution sensors and the use of *Object-Based Image Analysis* (OBIA) provide important means for the identification of urban

targets (Blaschke, 2010). OBIA can be regarded as an improvement of traditional pixel-based analysis techniques, specially when applied to high resolution spatial imagery. It defines image segments as analysis units, which can be characterized by a large number of spectral, morphological and topological features (Blaschke and Tomljenović, 2012). According to Francisco and Almeida (2012), a pixel does not meet the conceptual requirements of an “object” according to the OBIA paradigm, as does the segment, which can be characterized in such a way that it can conform to an interpretation model.

A considerable number of OBIA techniques has been successfully applied to urban planning, as demonstrated in the following examples.

Chen and Chen (2014) used WorldView-2 images for detecting changes in urban monitoring. Their results indicated that the object-based methodology significantly improved change detection accuracy as compared to pixel-based techniques. The global accuracy was close to 0.89, and the Kappa coefficient reached 0.65. Bias et al. (2014) used OBIA to evaluate the urban cadastre of Goianésia, Brazil. The authors used the InterIMAGE system together with the WEKA data mining package. Accuracy of the interpretation resulted in a Tau coefficient of 82.6. Orlando and La Rosa (2014) devised an object-based classification method to detect and analyze multi-temporal remote sensing data from Scopello, Italy. Using the eCognition software, the method reached an accuracy of 0.94 (kappa) in the detection of some of the classes of interest.

Due to the large variety of available features, specially in urban environments, object-based classification models, however, tend to be fairly complex and difficult to be designed solely based on empirical evidence or prior knowledge. According to Fayyad, Piatetsky-Shapiro and Smyth (1996), *Knowledge Discovery in Databases* (KDD) is the global process of discovering knowledge from data, and data mining is a specific step in the identification of patterns in the available data. Data mining techniques can, thus, be very helpful in the definition of interpretation models, making it possible to exploit the vast unequal potential of object features and to gain knowledge about specific characteristics of classes of objects.

The objective of this paper is to jointly use SIPINA and InterIMAGE, both free and open-source software packages, for the urban land cover object-based

classification of remotely sensed high spatial resolution data. The SIPINA contains implementations of various supervised learning algorithms, enabling interactive and visual construction of decision trees (Rakotomalala, 2008).

In this study we investigated the algorithms currently available in SIPINA (ID3, C45, GID3, ASSISTANT 86 and CHAID) and employed them in the design of classification models in InterIMAGE, an open-source, knowledge-based framework for automatic image interpretation.

STUDY AREA, MATERIALS AND METHODS

The study area is the municipality of Goianésia (Figura 1) in the State of Goiás, Brazil, located 168 km from Goiânia, the State's capital.

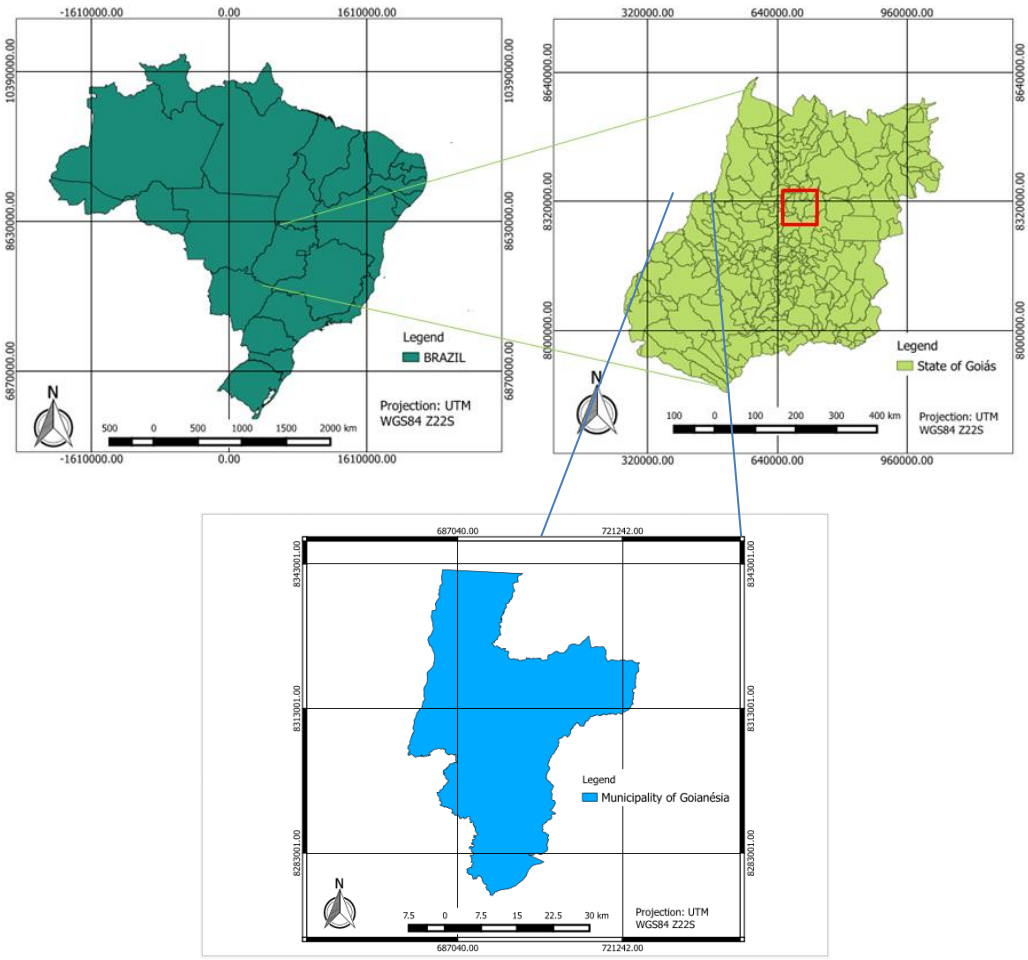


Figura 1 - Municipality of Goianésia, Goiás.

A pansharpened GeoEye-1 image acquired in 2013, covering the urban area of Goianésia, was used in this study. The image has a spatial resolution of 0.41 cm in the panchromatic band, and of 1.65 m in the multispectral bands (blue, green, red and near infrared).

The following open-source software packages were used in the study: QuantumGIS, version 2.10.1; SIPINA, version 3.12; and InterIMAGE, version 1.43.
QuantumGIS is a general purpose geoprocessing software, which contains tools for handling georeferenced images and vector data (QGIS BRASIL, 2015).
InterIMAGE was developed by researchers from the Catholic University of Rio de Janeiro (PUC-Rio) and from the Brazilian Space Research Institute (INPE), and encompass a set of methods for the design and implementation of object-based interpretation models (Costa et al., 2010).

SIPINA was developed at University of Lyon, France, and contains a set of specialized *Classification Trees* induction algorithms. The first version was distributed in 1995 (Kaur and Singh, 2013). It contains implementations of various supervised learning methods, enabling interactive and visual construction of classification trees (Rakotomalala, 2008).

As mentioned before, in this study we investigated a set of methods available in SIPINA: ID3, C45, GID3, ASSISTANT 86, and CHAID.

The ID3 algorithm (*Iterativo DiChaudomiser 3*) was originally developed by J. Ross Quinlan (1986) at the University of Sydney, Australia. The algorithm selects classification attributes for a decision tree based on entropy information and information gain. Entropy from Information Theory (the impurity of the attribute) is used to measure the information gain of an attribute. The information gain refers to the type of impurity. The lower the entropy value, the less uncertainty and more utility the pre-classified product has (Wilges et. al, 2010).

According to Hssina et al. (2014), the C4.5 algorithm was proposed in 1993, also by J. Ross Quinlan, to overcome the limitations of ID3, such as the sensitivity of resources in face of a high number of feature values.

The algorithm GID3 is a generalization of ID3 and C4.5, in which some leaves of the separation process may mix together. The idea is to highlight the more interesting leaves and merge the others into a “standard page” (Fayyad, 1994).

The Assistant 86 is another enhancement of ID3. In it there is a criteria set for improving the information gain and the various parameters which can control the size of the tree (Cestnik et al., 1987).

CHAID is an enhanced version of Morgan and Sonquist's (1963) AID algorithm. Kass (1980) explains the particularities of CHAID, such as the use of Chi-square statistics for division criteria, and fusing some pages into a single node.

At this point, it is important to note that the InterIMAGE package also includes a supervised classification operator that implements the C4.5 algorithm. Assuming that the implementation of the algorithm in InterIMAGE and in SPINA are similar, thus producing equivalent results, and bearing in mind that this particular study investigates the integration of the two open-source packages, we decided not to include it in a comparative evaluation, considering the C4.5 implementation built in InterIMAGE.

METHODOLOGY

The devised methodology has five steps: pre-processing; pre-classification; data mining; classification; and results analysis (Figura 2). The following subsections describe those steps.

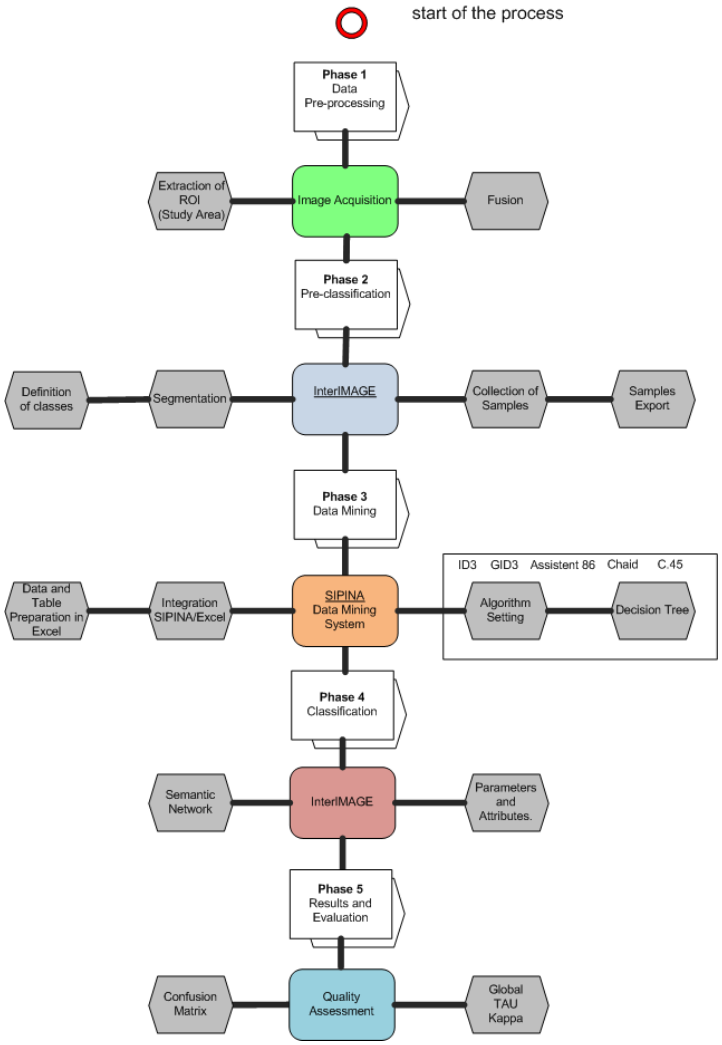


Figura 2 - Methodological steps.

Step 1: Data Pre-processing

In this work, we used a GeoEye-1 sensor image, from 2013, covering the city of Goianésia, Goiás, Brazil. The image was pansharpened and the ROI (Region of Interest) corresponding to the borders of the study area was extracted from the image.

Step 2: Pre-classification

Nine target classes were defined for the interpretation task: metallic roof; asbestos roof; ceramic roof (clear and dark); swimming pools; vegetation; bare soil; concrete pavement; and shadow.

After the definition of the classes of interest, image segmentation was performed in InterIMAGE, using the algorithm proposed by Baatz and Schäpe (2000). According to (Ferreira et al., 2013) the quality of the segmentation produced by that algorithm improves when the heterogeneity criteria that governs the growth of regions (segments) takes into account morphological attributes in addition to the spectral ones. According to the same authors, the quality gain can significantly depend on the characteristic shape of the objects of a particular class. In this work, therefore, the segmentation procedure was specialized for the different classes of interest, thus producing different segmentation outcomes. The segmentation parameters values used for the different classes are detailed in Table 1.

Classes	Input Band	Relative Band Weight	Compactness	Color	Scale
Metal roofing	0,1,2,3	1,1,1,1	0.8	0.4	90
Asbestos roofing	0,1,2,3	1,1,1,1	0.7	0.5	80
Clear ceramics roofing	0,1,2,3,	1,1,1,1	0.5	0.5	80
Dark Ceramics Roofing	0,1,2,3,	1,1,1,1	0.5	0.5	70
Swimming pools	0,1,2,3	1,1,1,1	0.8	0.4	60
Bare soil	0,1,2,3	1,1,1,1	0.5	0.5	60
Concrete pavement	0,1,2,3	1,1,1,1	0.5	0.5	60

Tabela 1 - Parameters used in image segmentation.

After image segmentation, samples from each class were collected using the InterIMAGE's *Sample Editor* tool. The Figura 3 shows the original image (a) and the corresponding segmentation for the metallic roof class (b), respectively. Examples of metallic roofs' sample objects are shown in (c).

In order to properly deal with the large variety of colors corresponding to metallic and asbestos roofs and to select samples for data mining (Step 3), these classes were divided into sub-classes: asbestos; asbestos_1; metallic; metallic_1; metallic_2; and metallic_3. The sub-classes were regrouped in the classification step (Step 4).

(a) (b) (c)

Figura 3 - Original image (a). Segmentation for metallic roofs (b). Samples of metallic roofs (c).

Following Antunes et al. (2014), image segmentation for the vegetation and shadow classes was performed using a per-pixel thresholding segmentation tool of InterIMAGE. The vegetation class thresholding was carried out using the Normalized Difference Vegetation Index (NDVI) feature. All contiguous pixels with NDVI above 0.65 were enclosed in the corresponding vegetation segments. The segmentation for the shadow class was based on pixel brightness values: brightness above 30 and below 54.

After segmentation, the following features were calculated for each segment (from the different segmentations):

a) Spectral features: compacity, brightness, entropy, maxpixelvalue, mean, minpixelvalue, ratio and bandmeandiv.

b) Morphological features: angle, squareness and circleness.

These attributes were used in the data mining process (Step 3).

Step 3: Data Mining

All segments' features were exported by InterIMAGE in *shapefile* format and then the corresponding data file was opened in Microsoft Excel with the SIPINA add-in (SIPINA.XLA) previously installed (Figura 4).

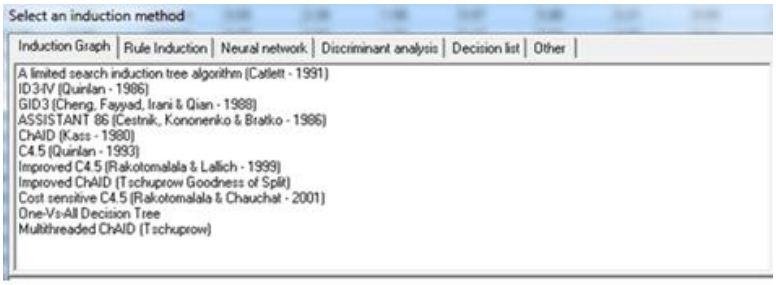


Figura 4 - Induction algorithms for SIPINA decision trees.

Then, decision trees were created using the methods ID3, C45, GID3, ASSISTANT 86, and CHAID.

Step 4: Classification

InterIMAGE was used for classification. An interpretation model in InterIMAGE contains information used by its control process to interpret a scene. It is represented by a semantic network in which the nodes are associated to classes of objects and are organized in a hierarchical fashion (Costa et al., 2010). The semantic network designed for this work is shown in Figura 5. The operators and respective used parameters are described in Bias et al. (2014).

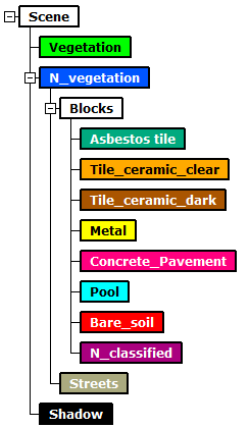


Figura 5 - Semantic network with defined classes.

The *TopDown Decision Rule* tool in InterIMAGE supports the creation of a set of expressions called decision rules. These expressions represent structured and specific knowledge used by the system in the interpretation (Costa et al., 2010). In

this work, the decision rules associated to each semantic network node were based on the decision trees generated automatically by SIPINA.

SIPINA is a specific data miner software for decision tree classification. The threshold values and the decision rules are defined by decision tree induction algorithms. According to Tedesco et al. (2014), the main objective of decision tree algorithms is to find the smallest possible decision tree, coherent with the training samples, achieving the correct classification with a small number of tests.

The automatically generated decision tree allows the analyst to inspect and study the tree structure (classes, values, attributes, and rules). In this way, it is possible for the analyst to identify in the tree the absence of attributes or classes due to their irrelevance (not useful in the classification process), or even due to operational error (when a given attribute or class is accidentally left apart).

Step 5: Analysis of the results

Five classification processes were conducted, based on the different decision tree induction algorithms in SIPINA.

The accuracy analysis was conducted on a test area based on the following evaluations established by Bias et al. (2014): measuring the number of samples; random distribution of check points; visual investigation; composition of the confusion matrix; calculation of the global accuracy, and the TAU and Kappa agreement coefficients. The number of samples were calculated by multinomial distribution. The sample unit for accuracy assessment was a pixel.

The amount of samples was determined according to Congalton and Green (1999), which was the same, used for the same area, as in Bias et al. (2014) and Antunes et al. (2014).

Afterwards, randomly generated check points (pixels) were determined for accuracy assessment. Each of the check points was visually inspected and assigned to its correspondent class. The automatic classification result was then compared to the visually assigned class for the construction of confusion matrixes and for calculating the global accuracy, TAU and Kappa agreement coefficients. The results are presented and discussed below.

The basic configuration of SIPINA was maintained for the decision tree induction algorithms, without altering any of the standard parameters.

The Figura 6 shows a decision tree generated using the algorithm **ID3**. The tree has 31 nodes, 16 leaves and a maximum depth of 6. Visualizing the tree eased the analysis of each class (Figura 43a). Each tree node shows a confidence percentage and the number of collected samples, seen in Figura 43b. Figuras 43c and 43d show the values for thresholds and features, respectively. The threshold values show the separation between two classes and the features on the tree are the ones SIPINA determined as having the best values.

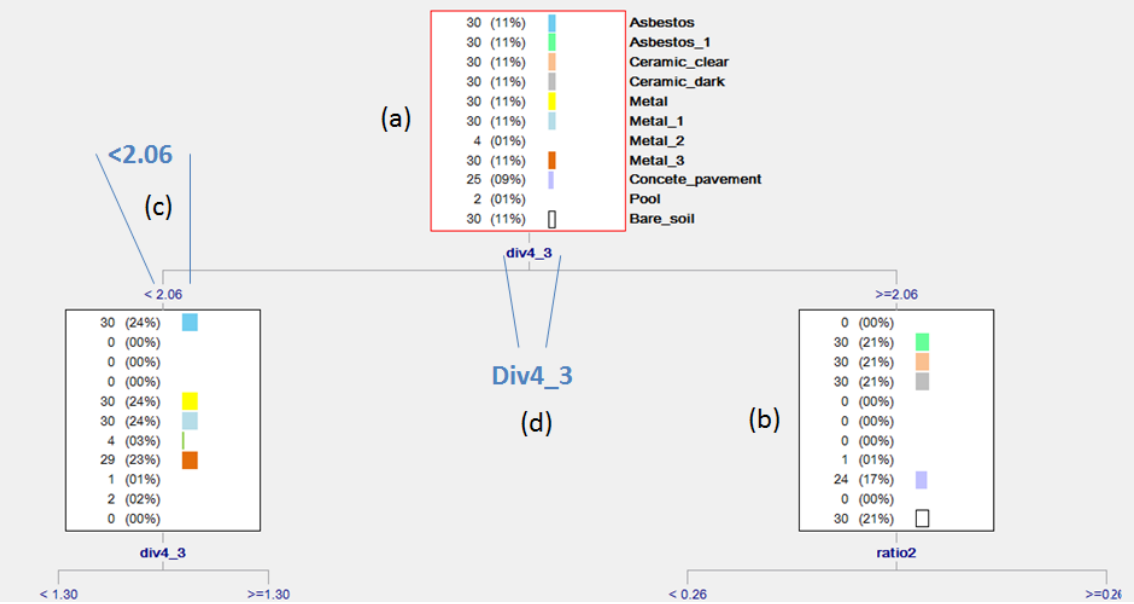


Figura 6 - Decision tree with decision rules generated by SIPINA using algorithm ID3. (a) Tree nodes with classes. (b) Confidence percentages and class samples. (c) Threshold value. (d) Attribute.

A confidence index of 50% was established for the rules contained in the leaves and chosen for insertion into InterIMAGE, as the rules with a low confidence index tend to reduce classification accuracy. According to Goldschmidt and Passos (2005), the measure of confidence expresses the quality of a rule. The example here is for the swimming pool class: the confidence index of the rule in the leaf reached 33%, as seen in Figura 7. The Figura also shows not-null confidence indices for other classes (metallic_1, metallic_2 and metallic_3). The rules associated to this leaf were ignored (not inserted in InterIMAGE) due to their low confidence index.

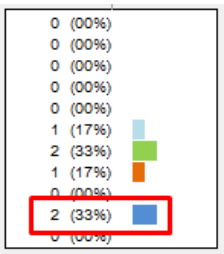
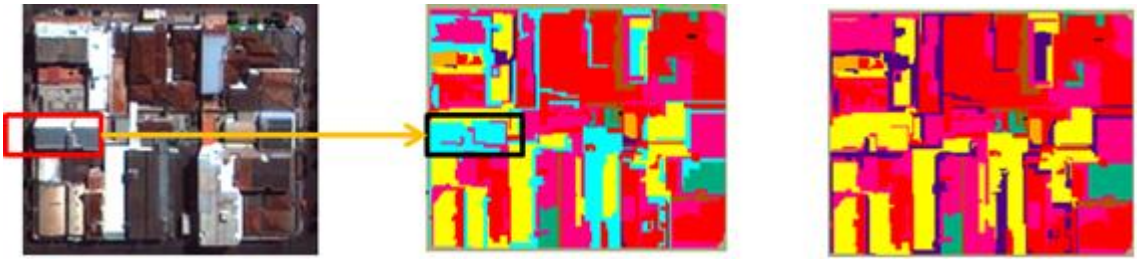


Figura 7 - Four classes (Metallic_1, Metallic_2, Metallic_3 and Swimming pool) with the same rule in the leaf and confidence ranging from 17% to 33%.

The swimming pool class rule, with a 33% confidence index, was inserted in InterIMAGE only to illustrate the confusion process generated in the classification. Figura 8 shows the questionable swimming pool classification generated by this rule. In (a) the original image of the block is shown without any Swimming pools. In (b) one can see the inaccurate classification of some objects as swimming pool (in cyan), with roofs classified as such. (c). Lastly, (d) shows the same classification with the swimming pool class rule discarted.



a.Original image of the block without swimming pool.

b. Insertion of the rule for the Swimming pool class (33% confidence index).

d. Classification disregarding the Swimming pool class rule.

Figura 8 - Result of a classification with a 33% confidence rule defined in SIPINA and executed in InterIMAGE.

The Table 2 shows an analysis of the decision tree based on the best rules using algorithm ID3. The rule for identifying the metallic_3 class was the most complex one, depending on three different criteria. There were also rejected rules for this class (two), which is related to those presenting a confidence index under 50%. The same dificulty was observed for the other tested algorithms.

Classes	Samples	Best rule	Confidence (Leaf)	Rejected rules (-50% confidence)
Asbestos roof	30	IF div4_3 < 2.06 and ratio3 >=0.20 and div4_1 >=2.00 and brightness < 993.93	100%	NAO
Asbestos roof_1	30	IF div4_3 >=2.06 and ratio2 >=0.26 and mean1 >=266.58 and ratio3 < 0.15 and minpixel1 >=226.00	100%	NAO
Ceramic_clear	30	IF div4_3 >=2.06 and ratio2 < 0.26 and mean3 >=317.63 and ratio2 < 0.23	100%	SIM (40%)
Ceramic_dark	30	IF div4_3 >=2.06 and ratio2 >=0.26 and mean1 < 266.58	100%	NAO
Metallic	30	IF div4_3 < 2.06 and ratio3 >=0.20 and div4_1 < 2.00	100%	NAO
Metallic_1	30	IF div4_3 < 2.06 and ratio3 < 0.20 and ratio1 >=0.22 and compacity < 0.07	100%	SIM (17%)
Metalica_2	4	IF div4_3 < 2.06 and ratio3 < 0.20 and ratio1 < 0.22 and brightness >=916.20	100%	SIM (33%)
Metalica_3	30	IF div4_3 < 2.06 and ratio3 < 0.20 and ratio1 < 0.22 and brightness < 916.20 and ratio3 < 0.17 IF div4_3 < 2.06 and ratio3 < 0.20 and ratio1 < 0.22 and brightness < 916.20 and ratio3 >=0.17 IF div4_3 < 2.06 and ratio3 >=0.20 and div4_1 >=2.00 and brightness >=993.93	50% 100% 100%	SIM (17% e 20%)
Pool	2	IF div4_3 < 2.06 and ratio3 < 0.20 and ratio1 >=0.22 and compacity >=0.07	33%	SIM (33%)
Bare soil	30	IF div4_3 >=2.06 and ratio2 < 0.26 and mean3 >=317.63 and ratio2 >=0.23 IF div4_3 >=2.06 and ratio2 < 0.26 and mean3 < 317.63	60% 100%	SIM (40%)
Concrete pavement	25	IF div4_3 >=2.06 and ratio2 >=0.26 and mean1 >=266.58 and ratio3 < 0.15 and minpixel1 < 226.00 IF div4_3 >=2.06 and ratio2 >=0.26 and mean1 >=266.58 and ratio3 >=0.15 and maxpixel2 < 743.50	100% 100%	SIM (40%)

Table 2 - Basic decision tree statistics with algorithm ID3.

The Table 3 and Figura 9 show the performance of each of the analyzed SIPINA algorithms.

Algorithms	Nodes	Leaf	Depth	Rejected rules (-50% confidence)
ID3	31	16	6	8
C45	27	14	7	7
GID3	31	16	7	5
Assistent 86	21	11	6	12
CHAID	39	20	8	4

Table 3 - Performance summary of each SIPINA decision tree algorithm.

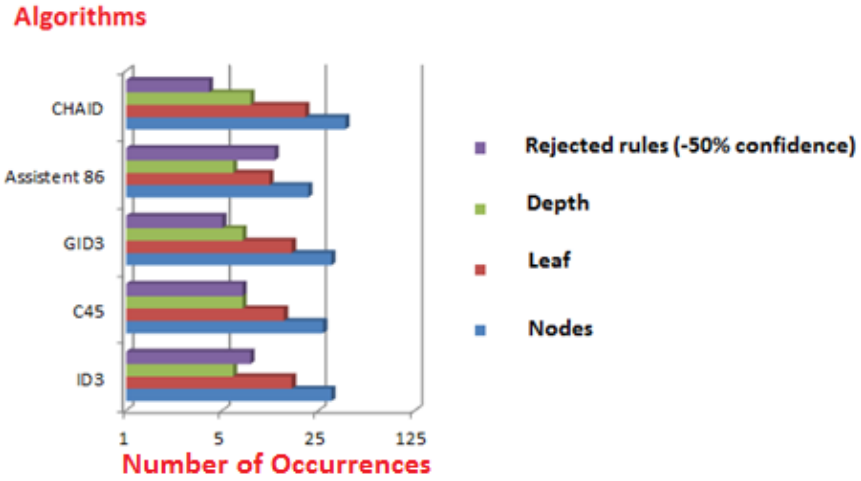


Figura 9 - Performance summary of each SIPINA algorithm.

As Table 3 and Figura 9 show, the algorithms presented a very little variation with respect to the corresponding tree structures. The CHAID algorithm produced the highest number of nodes (39), leaves (20) and and greatest structural depth (8), yet the number of rejected rules was low (4).

The tree with the most compact structure was produced by the Assistant 86 algorithm. It had 21 nodes, 11 leaves and a maximum depth of 6, although with a high number of rejected rules (12).

SIPINA did not have any difficulties processing decision trees for the data considered in this study, but its limitation of 16,384 attributes and 500,000,000 registers requires attention. This limitation has to do with the fact that the system loads the whole data set to the memory before the learning process begins (Rakotomalala, 2016).

As mentioned before, the decision rules with threshold values defined by the SIPINA algorithms were inserted into InterIMAGE using the *TopDown Decision Rule*. This process was performed manually. The Figura 10 shows an example of the rule insertion for the dark ceramic roof class.

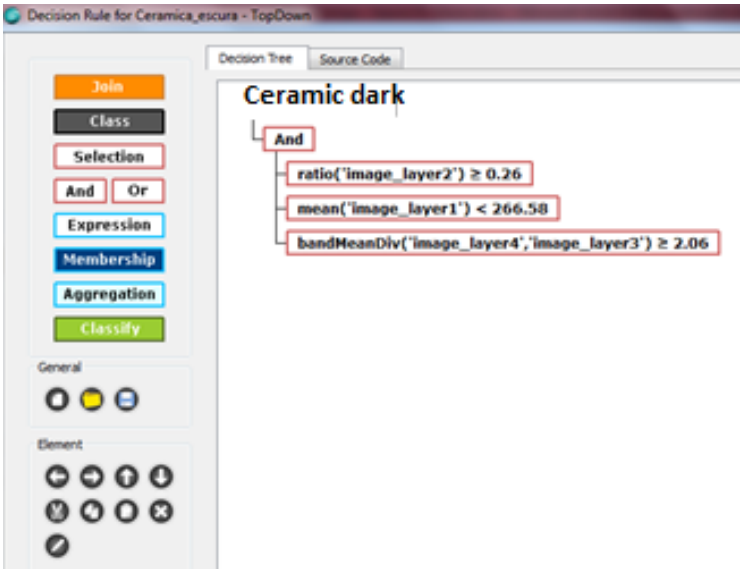


Figura 10 - Example of a decision rule defined in SIPINA and inserted in InterIMAGE.

Image classification consists of separated segment sets that exhibit similar characteristics (e.g., spectral, morphological or textural). The classification result is a thematic map showing the geographical distribution of the classes (Tedesco et al., 2014).

The Figura 11 visually shows each classification resulting from the SIPINA and InterIMAGE integration.

As expected, the confusion matrix derived coefficients, for each classification, showed greater confusion between ceramic roofs and bare soil classes, as they are composed of the same material (red clay): the spectral response also has a major influence in this respect. The vegetation and street classes were well classified, achieving a good separation, without much confusion.

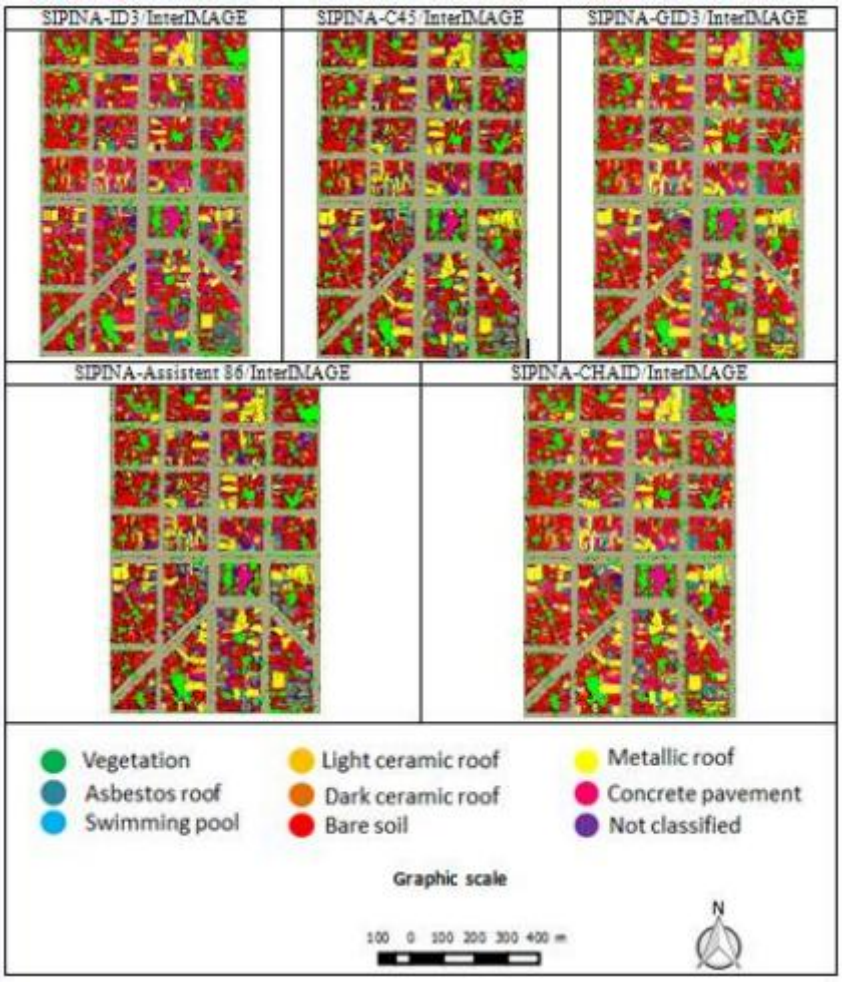


Figura 11- Results from each classification. SIPINA-InterIMAGE integration.

The quality of each classification was calculated by means of the global accuracy, the TAU and Kappa agreement coefficients. The Table 4 shows the corresponding accuracy values.

Algorithm	Global accuracy (%)	TAU	Kappa
ID3	70	0.66	0.65
C45	72	0.68	0.67
GID3	73	0.70	0.69
Assistent 86	73	0.70	0.69
CHAID	73	0.70	0.69

Table 4 - Global accuracy and TAU and Kappa agreement coefficients for the obtained classification results.

As Table 4 shows, classification using ID3 algorithm had the worst agreement coefficients (TAU 0.66 and Kappa 0.65). The GID3, Assistant 86 and CHAID algorithms all reached the same agreement indices (TAU 0.70 and Kappa 0.69).

CONCLUSION

The results obtained in this study allowed us to evaluate the integrated use of the SIPINA data mining package and the InterIMAGE system in an object-based image analysis application. The investigation leaded to the following conclusions:

a) SIPINA proved to be an easy-to-use software, working directly in Excel spreadsheets (without the need of installing other applets) and providing ways for visual analysis of decision trees and corresponding rules confidence values associated to each tree node, thus allowing the analyst to inspect the credibility of rules for each class of interest.

b) SIPINA offers a number of algorithms for decision tree induction: ID3, C4.5, GID3, Assistant86 and CHAID. With the exception of CHAID, from Morgan and Sonquist (1963), all the other algorithms are based on the Quinlan seminal algorithm.

c) The TAU and Kappa coefficients obtained with the GID3, Assistant86 and CHAID algorithms, 0.70 and 0.69, respectively, represent satisfactory classifications, not excellent ones, however. The confusion between bare soil and ceramic roof was, for the most part, responsible for reducing the values of such coefficients. Antunes et al. (2015) used the same input data, but employed the J4.8 algorithm implemented in the WEKA package, and obtained a TAU of 0.78 for the same classification. The J4.8 algorithm is a Java implementation of the Ross Quinlan's C4.5 algorithm. In this project, using the same input data and the C4.5 algorithm, the TAU index of 0.72 was attained. The presented results are very close, showing a small difference (0.6). Several factors may influence this difference though, such as internal characteristics of the data mining softwares (WEKA and SIPINA), different random samplings and others.

d) Among the algorithms tested in this work, Assistant 86 was more suitable for integration with InterIMAGE (at least for this particular classification problem). It reached the same agreement coefficient as the CHAID and GID3 algorithms, but the tree structure was more compact, having less nodes, leaves and depth. This means there were fewer rules inserted in InterIMAGE for classification, implying that the integration does not require much effort on the part of the analyst. Aditionally, there is a smaller chance of error in its operation and greater flexibility to interpret the rules translated into InterIMAGE.

ACKNOWLEDGEMENTS

The authors acknowledge the support provided by CAPES (Coordenação de Aperfeiçoamento de Pessoal de Nível Superior).

BIBLIOGRAPHIC REFERENCES

Antunes, R. R., Bias, E. S., Brites, R. S., Costa, G. A. O. P. Desenvolvimento de técnica para monitoramento do cadastro urbano baseado na classificação orientada a objetos. Estudo de caso: Município de Goianésia, Goiás. Revista Brasileira de Cartografia (2015) N0 67/2: 357-372. Brasília, Distrito Federal, 2014.

Baatz, M., Schäpe, A. Multiresolution segmentation: an optimization approach for high quality multi-scale image segmentation. In: XII Angewandte Geographische Informationsverarbeitung, AGIT Symposium. Proceedings. Karlsruhe, Alemanha: Herbert Wichmann Verlag, Salzburg - Áustria, p. 12-23, 2000.

Bias, E.S., Antunes, R.R., Pereira, E., Costa, G.A.O.P, Brites, R.S., Rithter, M. Application of Imagery Analysis Based on Objects as a Tool for Monitoring the Urban Cadastre in Small Municipalities. International Geographic Object-Based Image Analysis Conference, Thessaloniki, 2014.

Blaschke, T. Object based image analysis for remote sensing. Journal of Photogrammetry and Remote Sensing, Falls Church, v. 65, n. 1, p. 2–16, 2010.
Blaschke, T.; Tomljenovic, I. LidarScapes and OBIA. In Proceedings of the ASPRS 2012 Annual Conference, Sacramento, CA, USA, 19–23 March 2012.

Cerqueira, J. A. C., Alves, A. O. Classificação de imagens de alta resolução espacial para o mapeamento do tipo de pavimento urbano**.** III Simpósio Brasileiro de Ciências Geodésica e Tecnologias da Geoinformação. Recife, PE, 2010.

Cestnik, B., Kononenko I., Bratko I., " ASSISTANT 86: A Knowledge Elicitation Tool for Sophistical Users ", Proc. of the 2nd European Working Session on Learning, pp.31-45, 1987.

Chen, Q., Chen, Y. Object-based Change Detection of WorldView-2 data for Urban Dynamic Monitoring. South-Eastern European Journal of Earth Observation and Geomatics. Aristotle University of Thessaloniki, Greece. Vo3, No2S, 2014.

Congalton, R. G.; Green, K. Assessing the accuracy of remotely sensed data: principles and practices. Boca Raton-USA: Lewis Publisher, 1999.

Costa, G.A.O.P. ; Feitosa, R.Q. ; Fonseca, L.M.G. ; Oliveira, D.A.B. ; Ferreira, R.S. ; Castejon, E.F. Knowledge-based Interpretation of Remote Sensing Data With the InterIMAGE System: Major Characteristics and Recent Developments. GEOBIA 2010. Gent, Belgium, 2010.

Fayyad, U., Piatetsky-Shapiro, G., Smyth, P. From Data Mining to Knowledge Discovery in Databases. American Association for Artificial Intelligence. AI magazine 17.3 (1996): 37.

Fayyad, U. M. Branching on attribute values in decision tree generation. California Institute of Technology. Pasadena, CA. AAAI (www.aaai.org), 1994.

Ferreira, R.S., Costa, G.A.O.P., Feitosa, R.Q. Avaliação de critérios de heterogeneidade baseados em atributos morfológicos para segmentação de imagens por crescimento de regiões. Boletim de Ciências Geodésicas, sec. Artigos, Curitiba, v. 19, n 3, p.452-471, 2013.

Francisco, C.N., Almeida, C.M. Avaliação de desempenho de atributos estatísticos e texturais em uma classificação de cobertura da terra baseada em objeto. Boletim. Ciências. Geodésicas. vol.18 no.2 Curitiba, Paraná, Brazil, 2012.

Goldschmidt, R., Passos, E. Data Mining: Um Guia Prático. Elsevier. Rio de Janeiro, 2005.

Hssina, B., Merbouha, A., Ezzikouri, H. Erritali, M. A comparative study of decision tree ID3 and C4.5. International Journal of Advanced Computer Science and Applications, Special Issue on Advances in Vehicular Ad Hoc Networking and Applications. TIAD laboratory, Computer Sciences Department, Faculty of sciences and techniques. Sultan Moulay Slimane University, Morocco, 2014.

Kass G. An exploratory technique for investigating large quantities of categorical data. Applied Statistics, 29(2), pp. 119-127, 1980.

Kaur, A., Singh, S. Classification and Selection of Best Saving Service for Potential Investors using Decision Tree – Data Mining Algorithms. International Journal of Engineering and Advanced Technology. Vol-2, Issue-4, April 2013.

Morgan, J. N., Sonquist, J. A. Problems in the analysis of survey data, and a proposal. Journal of the American Statistical Association 58, 415-434. 1963.

Orlando, P., La Rosa, E. Object oriented methodology for change detection technique: the case of Scopello-Silicy. South-Eastern European Journal of Earth Observation and Geomatics. Aristotle University of Thessaloniki, Greece. Vo3, No2S, 2014.

Otukei Jr., Blaschke T. Land cover change assessment using decision trees, support vector machines and maximum likelihood classification algorithms. International Journal of Applied Earth Observation and Geoinformation. Elsevier. 2010.

QGIS Brasil. Comunidade de usuários QGIS Brasil. 2015. Available at: http://qgisbrasil.org/. Access on: 08/11/2015.

Quilan, J. “Induction of Decision Trees ", in Machine Learning, pp.81-106, 1986.

Rakotomalala, R. Introduction of a Decision Tree using SIPINA. Tutorial. Departamento de Informática e Estatística. University Lyon, France. 2008.

Rakotomalala, R. SIPINA Overview. Departamento de Informática e Estatística. University Lyon, France. Available at: http://eric.univ-lyon2.fr/~ricco/sipina.html. Access on: Feb. 23, 2016.

Rufino, I. A. A., Silva, S. T. Análise das relações entre dinâmica populacional, clima e vetores de mudança no semiárido brasileiro: uma abordagem metodológica. Boletim de Ciências Geodésicas, sec. Artigos, Curitiba, v. 23, no1, p.166 - 181, 2017.

Wilges, B., Mateus, G., Nassar, S., Bastos, R. Avaliação da aprendizagem por meio de lógica de fuzzy validado por uma Árvore de Decisão ID3. Novas Tecnologias na Educação. Centro interdisciplinar de novas tecnologias na educação – CINTED. Universidade Federal do Rio Grande do Sul – UFRGS. V. 8 Nº 3, December, 2010.

Witten, I. H., Frank, E., Hall, M. A., and Pal, C. J. Data Mining: Practical machine learning tools and techniques. Morgan Kaufmann, 2016.
Tedesco, A., Antunes, A. F. B., Oliani, L. O. Detecção de formação erosiva (voçoroca) por meio de classificação hierárquica e por árvore de decisão. Boletim Ciências Geodésicas, Curitiba, Paraná, Brazil. v. 20, n. 4, p.1005-1026, out-dez, 2014.

3 - ARTIGO 2

ANÁLISE DE SISTEMAS MINERADORES DE DADOS OPEN-SOURCE E SEUS ALGORITMOS DE CLASSIFICAÇÃO DE ÁRVORE DE DECISÃO INTEGRADOS COM O SISTEMA DE CLASSIFICAÇÃO BASEADA EM OBJETOS INTERIMAGE

Analysis of open-source data mining systems and its decision tree classification algorithms integrated with the object-oriented classification system InterIMAGE

Rodrigo Rodrigues Antunes[1]
Edilson de Souza Bias[1]
Ricardo Seixas Brites[1]
Gilson A. O. P. Costa[2]

[1]Universidade de Brasília – UnB
Instituto de Geociências - IG
Caixa Postal 04465 – Brasília-DF, Brasil.
rodrigorantunes@hotmail.com
edbias@unb.br
brites.ricado@gmail.com

[2]Universidade do Estado do Rio de Janeiro – UERJ
Instituto de Matemática e Estatística
R. São Francisco Xavier, 524, Maracanã – Rio de Janeiro-RJ, 20550-900, Brasil.
gilson.costa@ime.uerj.br

RESUMO

O objetivo principal da pesquisa foi avaliar a integração de vários sistemas mineradores de dados *open source* com o sistema de classificação baseado em objetos InterIMAGE. Os sistemas mineradores utilizados nesta pesquisa foram: SIPINA, RapidMiner Studio, KNIME Analytics Platform, Orange Canvas e WEKA. Para este estudo foi utilizada a imagem do sensor GeoEye-1, de 2013, da área urbana do município de Goianésia, Goiás, Brasil. O sistema InterIMAGE possibilitou a segmentação multiresolução e a coleta de amostras (com os respectivos atributos) de cada classe definida. Os resultados foram avaliados por meio da métrica de acurácia Kappa, demonstrando-se por meio de bons resultados a viabilidade de integração do sistema InterIMAGE com os mineradores utilizados. Pelos resultados ficou demonstrado que a quantidade de regras influencia diretamente o tempo de processamento da classificação no InterIMAGE mas não no resultado (acurácia da classificação). O resultado de cada classificação foi satisfatório, de acordo com o índice Kappa de cada classificação: SIPINA-ID3 (66% Kappa), SIPINA-C45 (68% Kappa), SIPINA-GID3 (70% Kappa), SIPINA-Assistent86 (70% Kappa), SIPINA-CHAID (70% Kappa), RapidMiner (77% K0appa), KNIME (73% Kappa) Orange Canvas (81% Kappa) e WEKA (78% Kappa). Além disso, para analisar a significância de cada resultado da classificação por meio dos índices utilizou-se um teste estatístico (teste Z).

Palavras-chave: OBIA, InterIMAGE, Data Mining, Kappa, teste z.

ABSTRACT

The main objective of the research was to evaluate the integration of several open-source data mining software packages with the object-based classification system InterIMAGE. The data mining systems used in this research were: SIPINA, RapidMiner Studio, KNIME Analytics Platform, Orange Canvas and WEKA. We used a GeoEye-1 sensor image from 2013, covering the urban area of the city of Goianésia, Goiás, Brazil. The open source system InterIMAGE was used for the multiresolution segmentation and sample collection, for each class of interest. The results were evaluated through the comparisson of the Kappa Index, and demonstrate the viability of the integration of the data mining systems with InterIMAGE. The results showed that the total number of rules directly influenced the classification processing time in InterIMAGE but not in the quality of the results (accuracy of classification). The result of each classification was satisfactory, according to the Kappa accuracy index: SIPINA-ID3 (66% Kappa), SIPINA-C45 (68% Kappa), SIPINA-GID3 (70% Kappa), SIPINA-CHAID (70% Kappa), RapidMiner (77% Kappa), KNIME (73% Kappa) Orange Canvas (81% Kappa) and WEKA (78% Kappa). In order to analyze the significance of each classification result by means of the Kappa indices a statistical test (Z test) was carried out.

Keywords: OBIA, InterIMAGE, Data Mining, Kappa, z test.

INTRODUÇÃO

Um dos principais usos de imagens de sensoriamento remoto (SR) é a extração de informações de sobre a cobertura da terra (GRIPPA et al., 2016). Do ponto de vista científico, para se extrair informações de imagens de SR é necessário estabelecer métodos de análise com regras claras e coerentes, que possam ser reproduzidas por qualquer analista no processo de interpretação. Essas regras estabelecem critérios para a análise dos elementos que compõem a paisagem, e se baseiam em propriedades como cor, tonalidade, textura, estrutura e homologia (MENESES; SANO, 2012).

A evolução dos sensores de alta resolução espacial expôs as limitações das técnicas tradicionais de classificação pixel-a-pixel (BLASCHKE, 2001). A interpretação de imagens por procedimentos clássicos baseados em pixels, como a classificação multiespectral supervisionada, fornece geralmente resultados fragmentados e

insatisfatórios para imagens de alta resolução (LEUKERT, 2007).

Vários dos aspectos que envolvem a análise de imagens de SR, notadamente de imagens de alta resolução espacial, não podem considerados utilizando-se apenas informações provenientes de pixels individuais, uma vez que estas não representam o contexto geográfico (BLASCHKE; BURNETT; PEKKARINEN, 2004). A técnica de OBIA – Object Based Image Analysis permite a inclusão de informações adicionais nos processos de classificação e modelagem, relativas a segmentos, ou objetos de imagens, que são agrupamentos contíguos de pixels. Essas informações podem estar relacionadas com estatísticas sobre os conjuntos de pixels que compõem os objetos, sobre a textura e forma dos objetos, ou sobre relações topológicas entre objetos (BLASCHKE, 2013).

Métodos baseados em OBIA possuem geralmente três etapas principais: segmentação de imagem; extração de atributos, ou propriedades dos segmentos; e classificação. A segmentação de imagem é um procedimento para particionar uma área da imagem inteira em segmentos (ou objetos de imagem), que são grupos de pixels com valores espectrais homogêneos. Após a segmentação e análise de segmentos representativos de cada classe de objetos de interesse, a classificação é realizada com base em segmentos de imagem. Em OBIA, a qualidade da segmentação influencia diretamente o resultado da classificação de uma imagem SR (BLASCHKE, 2003; DORREN et al., 2003; MEINEL; NEUBERT, 2004; ADDINK et al., 2007).

Uma ferramenta de destaque no contexto de classificação de imagens de alta resolução espacial é o sistema InterIMAGE, plataforma livre baseada em conhecimento para interpretação automática de imagens (COSTA et al., 2010), que possui operadores para aplicar a OBIA no processo de classificação. Entretanto, é importante ressaltar que versão atual do InterIMAGE possui limitação para processar e classificar imagem superior a 9 Megapixels (3.000 x 3.000 pixels), e uma nova plataforma, chamada InterCloud – InterIMAGE Cloud Platform, uma reformulação do atual sistema concebida para processar grandes volume de dados em clusters de computadores físicos ou virtuais, vem sendo desenvolvida (FERREIRA et al., 2017).

No processo de interpretação de imagem, executado pelo sistema InterIMAGE, a definição de limiares para uma regra de decisão é importante para que se atinja uma boa acurácia na classificação. Estas regras podem ser definidas contemplando-se, além de valores espectrais, propriedades texturais,

morfológicas e topológicas de objetos de imagens.

A utilização do InterIMAGE integrada com um sistema de mineração de dados, para auxiliar a definição de regras e valores de limiares, tem apresentado bons resultados até o momento, sendo que, em diversos trabalhos publicados até o momento, o sistema minerador WEKA (GHOLAP, 2012) foi utilizado. Neste contexto, é interessante testar e avaliar a integração de outros sistemas mineradores de dados e seus algoritmos de classificação com o InterIMAGE de modo a disponibilizar ao analista alternativas de mineradores eficientes para o desenvolvimento de projetos.

Otukei e Blaschke (2010) fizeram uma comparação entre várias técnicas de classificação de mineração de dados para avaliar alterações no uso do solo através de OBIA. Entre as técnicas, foram testadas: Árvore de Decisão; Máquinas de Vetores de Suporte; e Máxima Probabilidade. Os autores concluíram que o desempenho do processo de classificação foi melhor com a técnica de Árvore de Decisão.

Neste trabalho a técnica de classificação de Árvore de Decisão (BREIMAN et al., 1984) foi investigada no contexto de OBIA. Foram analisados diversos algoritmos, implementados nos seguintes pacotes para a mineração de dados: SIPINA, RAPIDMINER STUDIO, KNIME ANALYTICS PLATFORM, ORANGE CANVAS e WEKA. Os resultados produzidos pelos vários algoritmos, em termos de regras de decisão e valores de limiares, foram inseridos no sistema InterIMAGE, para a classificação baseada em objetos, e as acurácias das respectivas classificações foram analisadas.

A justificativa da presente pesquisa ocorre em função do software WEKA representar o único minerador de dados utilizado e identificado nos artigos que desenvolvem aplicações com OBIA, em combinação com o sistema InterIMAGE, conforme pode-se observar nos trabalhos:

a) Análise de imagem baseada em objeto e mineração de dados aplicadas à classificação do uso do solo urbano por quadra em imagens WorldView-2 (CARVALHO et al., 2013);

b) Classificação de imagem baseada em objeto (OBIA) utilizando índices de vegetação (KAWASHIMAET et al., 2013);

c) Classificação da Cobertura da Terra, Utilizando os Programas Livres: InterIMAGE, WEKA e QuantumGIS (NASCIMENTO et al., 2013);

d) Geobia e mineração de dados na classificação da cobertura do solo urbano em São Luís (MA) com imagens Worldview-2 e o sistema InterIMAGE (SOUSA; KUX, 2014);

e) Desenvolvimento de técnica para monitoramento do cadastro urbano baseado na classificação orientada a

objetos. Estudo de caso: Município de Goianésia, Goiás (ANTUNES et al., 2014);

f) Técnicas de mineração de dados aplicadas à classificação do estádio sucessional da vegetação em áreas de floresta ombrófila mista (SOTHE et al., 2016);

g) Mapeamento da cobertura da terra do município de Raposa (Ma) utilizando imagens Worldview-II, o aplicativo InterIMAGE e mineração de dados (MENEGHETTI et al., 2014);

h) Aplicação de mineração de dados e técnicas GEOBIA para análise de susceptibilidade ao fogo no Parque Nacional Itatiaia, no Brasil (DE SOUSA et al., 2014);

i) Uso do sistema InterIMAGE para a identificação de alvos urbanos em imagens do satélite Worldview II (PASSO et al., 2013);

j) Classificação orientada a objeto em associação às ferramentas reflectância acumulada e mineração de dados (DE GRANDE et al., 2017);

k) Aplicação da Análise de Imagens Baseada em Objetos como Ferramenta de Monitoramento do Cadastro Urbano em Pequenos Municípios (BIAS et al., 2014); e

l) Análise do nível de legenda de classificação de áreas urbanas empregando imagens multiespectrais e hiperespectrais com os métodos árvore de decisão C4.5 e floresta randômica (DOS ANJOS et al., 2017).

Assim sendo, o objetivo do presente trabalho foi avaliar os principais pacotes de *software* mineradores de código aberto, além do WEKA, e seus algoritmos de indução de árvore de decisão, de forma a a avaliar a integração com o sistema InterIMAGE na classificação baseada em objetos de imagens de SR.

MATERIAIS

Dados da Imagem

A imagem de SR utilizada nos experimentos recobre uma área localizada no município de Goianésia (Figura 12), na parte centro-norte do estado de Goiás, Brasil.

Trata-se de uma imagem do sensor GeoEye-1, originalmente com uma banda pancromática, com resolução espacial de 50 cm, e quatro bandas multiespectrais (azul, verde, vermelho e infravermelho), cada uma com resolução espacial de 2 metros.

Nos experimentos utilizou-se um recorte fusionado (*pansharpened*) da região central da cidade de Goianésia, com 977 por 1531 pixels (ANTUNES et al., 2014).

Pacotes de Software

Os pacotes de *software* mineradores de dados utilizados nesta pesquisa foram:

(a) SIPINA

O SIPINA é um *software* livre e acadêmico desenvolvido pela Universidade de Lyon, França, voltado à mineração de dados e indicado para a indução de árvores de decisão (*Classification Trees*). Teve sua primeira versão distribuída em 1995 (KAUR; SINGH, 2013) e, de acordo com Rakotomalala (2008), o mesmo implementa vários algoritmos de aprendizado supervisionado, com uma construção interativa e visual de árvore de decisão. Os principais algoritmos de indução de árvore de decisão disponíveis e testados no SIPINA foram: ID3, C45, GID3, Assistent86 e CHAID.

O algoritmo ID3 (Iterativo DiChaudomiser 3) foi originalmente desenvolvido por J. Ross Quilan (1986), na Universidade de Sydney, Austrália. Nesse algoritmo, a escolha dos atributos a serem utilizados pela árvore se dá a partir de informações de entropia e ganho de informação. Para medir o ganho de informação de um atributo se utiliza o conceito de entropia, da Teoria da Informação. O valor da entropia corresponde à impureza do atributo, sendo o ganho de informação a variação da impureza. Quanto menor o valor da entropia, menor a incerteza e mais utilidade tem o produto para a classificação (WILGES et al., 2010).

O algoritmo C.45 foi proposto em 1993, novamente por Ross Quinlan, para superar as limitações do algoritmo ID3 como, por exemplo, a dificuldade de tratar um grande número de valores (HSSINA et al., 2014).

O algoritmo GID3 é uma generalização do ID3 e C45, de Quilan, onde algumas folhas de um processo de separação podem ser mescladas. A ideia é destacar as folhas mais interessantes (FAYYAD, 1994).

O algoritmo Assistent86 é outra melhoria do ID3, de Quilan. Nesse algoritmo há um critério de melhoria do ganho da informação e vários parâmetros que permitem controlar o tamanho da árvore (CESTNIK; KONONENKO; BRATKO, 1987).

O algoritmo CHAID é uma versão aprimorada do algoritmo AID de Morgan e Sonquist (1963). Segundo Kass (1980), as particularidades do CHAID são: (1) A utilização da estatística qui-quadrado como no critério de divisão; (2) a fusão de algumas folhas que vêm do mesmo nó. Este algoritmo é amplamente difundido em *software* comerciais. Por outro lado, é menos implementado em *software* livre.

(b) RAPIDMINER STUDIO

O minerador de Dados RapidMiner Studio, é um sistema de código aberto, desenvolvido na linguagem Java, para

concepção de processos analíticos avançados com aprendizagem de máquina, mineração de dados, mineração de texto, análise preditiva e análise de negócios (RAPIDMINER, 2017). A primeira versão do RapidMiner Studio surgiu em 2001 com nome de YALE (*Yet Another Learning Environment*), sistema desenvolvido pelo departamento de inteligência artificial da Universidade de Dortmund, da Alemanha (FAULHABER, 2007). A versão atual (7.0) do RapidMiner Studio permite desenvolver o processo de análise de dados por meio de vários operadores disponíveis em um ambiente de visualização gráfica.

(c) KNIME ANALYTICS PLATFORM

O KNIME Analytics Platform é um sistema de código aberto, baseado na plataforma Eclipse (Java) com possibilidade de customizar e implementar novos módulos em um ambiente visual (KNIME, 2017). Esta modularidade permite a aplicação do KNIME Analytics Platform em ambientes de produção comercial, ensino e pesquisa. O desenvolvimento do KNIME Analytics Platform começou em 2004, por um grupo de desenvolvedores na Universidade de Konstanz, Alemanha. O objetivo era atender demandas de processamento e análise de grande quantidade de dados de uma indústria farmacêutica. Em 2006, foi lançada a primeira versão deste minerador. Atualmente, encontra-se na versão 3.1.2. O *software* permite trabalhar de forma compartilhada e colaborativa (KNIME Team Space e KNIME Server Lite), suporta escalabilidade sob demanda em nuvem (KNIME Cloud Server), possui extensão para Big Data (Hadoop) (KNIME Big Data Extensions e KNIME Big Data Connectors) e pode ser executado em ambiente de processamento em cluster (KNIME Cluster Execution) (KNIME, 2017).

(d) ORANGE CANVAS

O sistema minerador *open source* Orange Canvas, foi desenvolvido na Linguagem Phyton. É um conjunto abrangente de *software* baseado em componentes para mineração de dados e aprendizagem de máquina, desenvolvido no Laboratório de Bioinformática, da Faculdade de Computação e Ciência da Informação da Universidade de Ljubljana, Eslovénia, juntamente com a comunidade de código aberto (DEMSAR; CURK; ERJAVEC, 2013). O Orange Canvas trabalha com blocos de construção de fluxos chamado *Widgets* para análise de dados que são montados no ambiente de programação visual do sistema. Os *Widgets* são agrupados em classes de acordo com sua função.

(e) WEKA

WEKA (Waikato Environment for Knowledge Analysis) é um *software* de mineração de dados de código aberto desenvolvido na linguagem Java. É desenvolvido na Universidade de Waikato (Nova Zelândia) e teve a sua primeira versão lançada em 1997. Este sistema possui um conjunto de algoritmos de aprendizado de máquina para tarefas de mineração de dados. Os algoritmos podem ser aplicados diretamente a um conjunto de dados ou chamado a partir de seu próprio código Java. Vários são os algoritmos de indução de árvore de decisão disponíveis WEKA, entre eles J48, REPTree, Decision Stump, RandomTree e outros. Para este trabalho foi testado e analisado o J48, que é a implementação do algoritmo em C4.5 no minerador WEKA (GHOLAP, 2012).

Para a segmentação e classificação baseada em objetos foi utilizado o sistema InterIMAGE (InterIMAGE, 2010).

Para a leitura de dados matriciais, vetoriais e confecção de mapas foi utilizado o *software* QuantumGIS, Versão 2.18.1 64 bits (QGIS BRASIL, 2017)

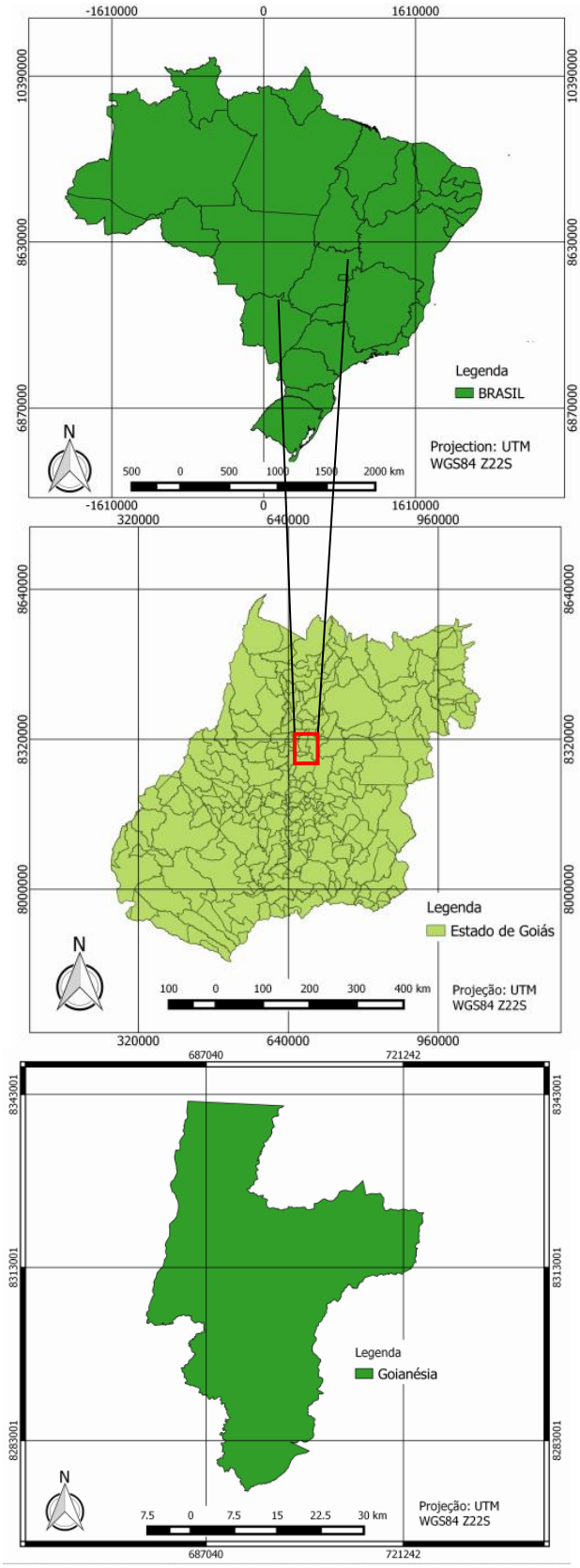


Figura 12 - Localização da área de estudo – Goianésia, Goiás, Brasil.

METODOLOGIA

A Figura 13 representa as etapas metodológicas cumpridas para alcançar os resultados da pesquisa.

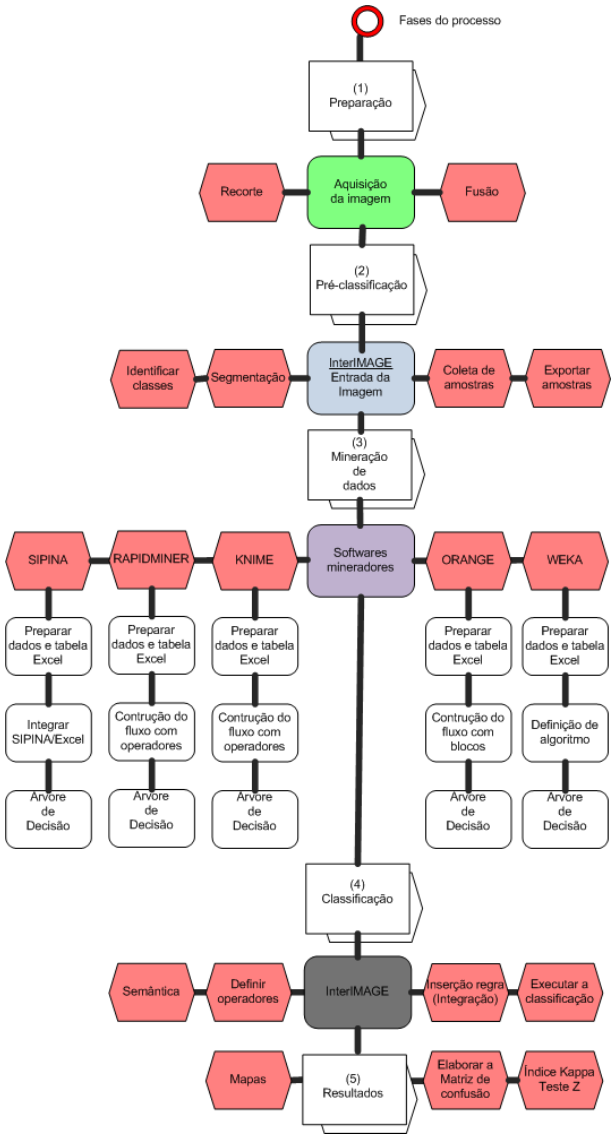


Figura 13 - Esquema das etapas do trabalho.

Fase (1): Preparação

A imagem original GeoEye-1 foi fusionada (*pansharpened*), utilizando-se o método GS – Gram-Schmidt. Posteriomente foi realizado recorte da área central, em uma dimensão de 977x1531 pixels.

Fase (2): pré-classificação

Nessa fase foram definidas as classes de interesse: telhado metálico, telhado de amianto, telhado de cerâmica clara, telhado de cerâmica escura, piscina, vegetação, solo exposto, piso de concreto, vias urbanas e sombra.

A segmentação da imagem foi desenvolvida por meio do operador TA_Baatz_Segmenter do InterIMAGE, adaptado de Baatz e Shape (1999), gerando diferentes segmentações para diferentes classes, de acordo com os parâmetros apresentados na Tabela 5. Após a segmentação, foram coletadas amostras de cada classe definida.

Table 5 - Parâmetros utilizados na segmentação

Classes / nº amostras	Banda de Entrada	Peso das Bandas	Compacidade	Cor	Escala
Metálico (25)	0,1,2,3	1,1,1,1	0.8	0.4	90
Amianto (25)	0,1,2,3	1,1,1,1	0.7	0.5	80
Cerâmica Clara (20)	0,1,2,3	1,1,1,1	0.5	0.5	80
Cerâmica Escura (20)	0,1,2,3	1,1,1,1	0.5	0.5	70
Piscina (30)	0,1,2,3	1,1,1,1	0.8	0.4	60
Solo Exposto (30)	0,1,2,3	1,1,1,1	0.5	0.5	60
Piso de concreto (25)	0,1,2,3	1,1,1,1	0.5	0.5	60

Para as classes não apresentadas na Tabela 5 (vegetação, vias urbanas e sombra) não foram coletadas amostras, pois foram utilizados os seguintes operadores do InterIMAGE: TA_NDVI_Segmenter para vegetação, TA_ShapeFile_Import para vias urbanas e TA_Arithmetic para sombra.

Fase (3): Mineração de Dados

Utilizando a planilha Excel, foi possível preparar os dados para leitura do

minerador (pré-processamento), permitindo iniciar o processo de mineração de dados com cada sistema minerador (SIPINA, RapidMiner Studio, KNIME Analytics Platform, Orange Canvas e WEKA).

Fase (4): Classificação - InterIMAGE

A classificação baseada em objetos, segundo Orlando e La Rosa (2014), representa a segmentação de uma imagem em objetos (grupos de pixels). Estes objetos têm características geográficas como forma, comprimento e entidades topológicas. Estes atributos formam uma base de conhecimento para os objetos, que podem ser chamados no processo de classificação.

O InterIMAGE permitiu a criação da rede semântica das classes de acordo com o objetivo do trabalho e a janela *TopDown* Decision permitiu criar um conjunto de expressões, denominadas regras de decisão. Estas expressões definem o conhecimento estruturado e explícito do analista e são utilizadas pelo sistema no processo de interpretação (InterIMAGE, 2010). De acordo com Costa et al. (2008), no InterIMAGE, o conhecimento explícito sobre os objetos, extraível a partir da própria cena, é modelado em uma rede semântica definida pelo usuário por meio de uma interface gráfica. A rede semântica é uma forma alternativa de representação de classe, que apresenta a maneira pela qual as diferentes categorias são conectadas com as respectivas classes de interesses e como estão espacialmente relacionadas com estas mesmas classes. Elas modelam e armazenam o conhecimento especializado (MAVRANTZA; ARGIALAS, 2008).

Um conjunto de operadores com funções é oferecido pelo InterIMAGE e, para este trabalho, foram utilizados: TA_Baatz_Segmenter, TA_Arithmetic, TA_ShapeFile_Import, Dummy TopDown e TA_NDVI_Segmenter (InterIMAGE Wiki, 2014).

Na Figura 14 é apresentada a rede semântica e o operador utilizado em cada classe.

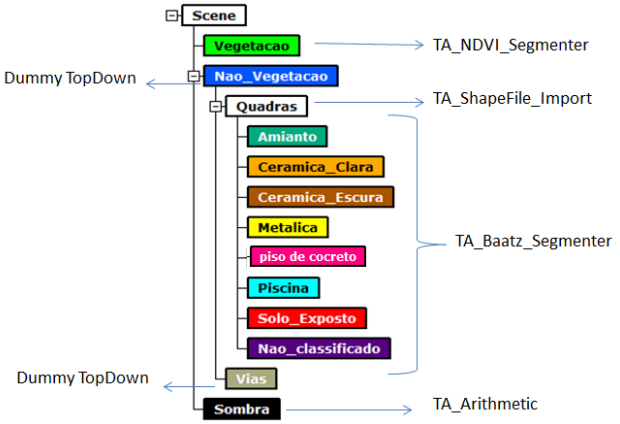


Figura 14 - Rede semântica e operadores utilizados no trabalho. Sistema classificador InterIMAGE.

No campo da Inteligência Artificial, a representação do conhecimento pode ser por meio de uma rede semântica, onde a informação é representada como um conjunto de nós conectados um ao outro por um conjunto de arcos rotulados que representam relações entre os nós (RICH, 1988). Para Bracham (1983), rede

semântica é um grafo rotulado e direcionado formado por um conjuto de nós representando os objetos e por um conjunto de arcos representando as relações entre os objetos.

A regra de decisão com valores de limiares definidos pelos algoritmos dos mineradores SIPINA, RapidMiner Studio, KNIME Analytics Platform, Orange Canvas e WEKA foram inseridos no classificador InterIMAGE por meio do operador *TopDown Decision Rule*, conforme ilustrado na Figura 15. Essa inserção de dados foi feita manualmente, o que representa a possibilidade da inserção de erro por parte do operador, sendo um dos fatores limitadores da utilização de mineradores de dados para o sistema InterIMAGE.

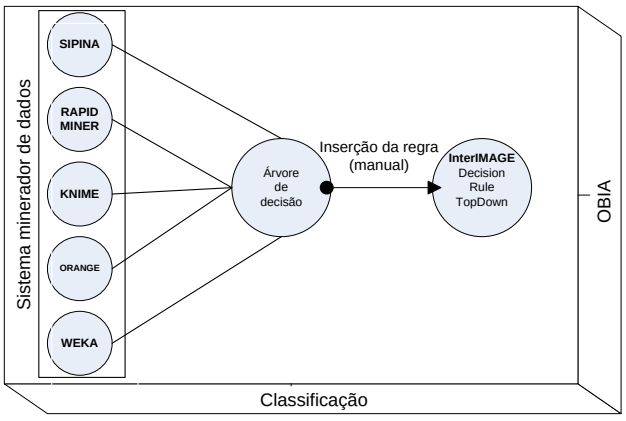


Figura 15 - Ilustração da integração dos mineradores SIPINA, RapidMiner Studio, KNIME Analytics Platform, Orange Canvas e WEKA com o InterIMAGE para classificação.

ANÁLISE DA QUALIDADE DAS CLASSIFICAÇÕES

A análise da acurácia temática foi aplicada seguindo as seguintes avaliações: quantificação do número de amostras, distribuição aleatória de pontos de checagem, investigação por interpretação visual, composição da matriz de confusão e coeficiente de concordância Kappa. Por meio de uma distribuição multinominal, foi calculado o número de amostras. A unidade amostral utilizada foi o pixel. Detalhes referentes aos procedimentos podem ser consultados em Passo et al. (2013) e Antunes et al. (2014).

De acordo com Cadena (2011), normalmente duas estratégias podem ser usadas no sentido de calcular o número de amostras necessárias para computar a acurácia: a distribuição binomial e a multinomial. O modelo binomial é apenas apropriado para computar o número de amostras necessário para uma única classe visto que este simplesmente faz a distinção entre classificação correta e incorreta e não leva em conta o conjunto de classes analisadas. Por outro lado, na distribuição multinomial, o processo de validação não é uma questão apenas de certo ou errado, mas o erro deve ser classificado em relação ao total de possibilidades (ou conjunto de classes), de modo que o uso dessa distribuição implica no conhecimento a priori do número de classes e suas proporções no mapa.

Segundo Congalton e Green (1999) a criação de uma matriz de erros não é simplesmente uma questão de correto ou incorreto (o caso binomial), e sim uma

questão de qual ou quais categorias de erros estão confusas. Portanto, o uso de uma distribuição binomial para a determinação do tamanho da amostra para uma matriz de confusão não é apropriado e sim uma distribuição multinomial, já que esta leva em consideração o número de classes.

Neste trabalho, após o processo de classificação e da geração de um mapa temático, foi criada, uma matriz de confusão para analisar a qualidade (testar a exatidão das aplicações dos diversos mineradores), aplicando-se o índice Kappa. Para analisar a significância de cada resultado da classificação utilizou-se um teste estatístico (teste Z).

A quantidade de amostras foi definida a partir da equação (1), que gerou um quantitativo de 664 amostras (CONGALTON; GREEN, 1999).

$$N = \frac{B\prod_i(1-\prod_i)}{b^2{}_i} \quad (1)$$

Onde:

N = número de amostras;
B = obtido da tabela de distribuição qui-quadrado;
1 Grau de liberdade e 1 - α / k;
$\prod_i$ = no mapa, a proporção da classe com maior ocorrência;
1-α = grau de confiança;
K = número de classes;
b = erro admissível.

O índice de acurácia Kappa (COHEN, 1960) é expresso de acordo com a equação (2).

$$K = \frac{P_o - P_c}{1 - P_c} \quad (2)$$

Po = Precisão Global (Proporção de unidades que concordam); e
Pc = Proporção de unidades que concordam por coincidência, representada pela equação (3):

$$P_c = \frac{\sum_{i-1}^{M} n_{i+} n_{+i}}{N^2} \quad (3)$$

M = número de classes;
ni+ = total de elementos classificados para categoria i;
n+i = total de elementos de referência amostrados para uma categoria i; e
N = número total de amostras.

De acordo com Amorim et al. (2016), para cada índice Kappa é possível calcular um intervalo de confiança por meio da variância da amostra (σ2) e o método comumente utilizado na comunidade do sensoriamento remoto é o teste “Z”.

A significância de um único índice tem como finalidade determinar se o nível de acerto da classificação e os dados de referência são significativamente maiores que zero. O teste feito para dois índices traduz se realmente existe diferença significativa entre os dois índices testados (CONGALTON; GREEN, 1999).

A variância da amostra de uma Matriz de Confusão pode ser calculada a partir do método de *Delta* (Amorim et al., 2016), de acordo com as equações equações : 4, 5, 6, 7, 8, 9 e 10.

$$\sigma_k^2 = \frac{1}{n}\left[\frac{\theta_1.(1-\theta_1)}{(1-\theta_2)^2} + \frac{2.(1-\theta_1).(2\theta_1\theta_2-\theta_3)}{(1-\theta_2)^3} + \frac{(1-\theta_1)^2.(\theta_4-4\theta_2{}^2)}{(1-\theta_2)^4}\right] \quad (4)$$

$$\theta_1 = \frac{1}{n}\sum_{i=1}^{c} X_{ii} \quad (5)$$

$$\theta_2 = \frac{1}{n^2}\sum_{i=1}^{c} X_{i+}X_{+i} \quad (6)$$

$$\theta_3 = \frac{1}{n^2}\sum_{i=1}^{c} X_{ii}(X_{i+} + X_{+i}) \quad (7)$$

$$\theta_4 = \frac{1}{n^3}\sum_{i=1}^{c}\sum_{j=1}^{c} X_{ij}(X_{j+} + X_{+j})^2 \quad (8)$$

Com a variância de todos os coeficientes Kappa calculada, os testes de significância podem ser efetivados utilizando as equações (9) e (10) que se seguem (AMORIM et al., 2016):

$$Z = \frac{k}{\sqrt{\sigma 2}} \quad (9)$$

$$Z_{1-2} = \frac{k_2 - k_1}{\sqrt{\sigma_{k2}^2 + \sigma_{k1}^2}} \quad (10)$$

O teste z para os índices Kappa das classificações foi realizado a 95% de significância. Quando z > 1,96, o teste é significativo, rejeita-se a hipótese de nulidade, podendo concluir que existe diferença estatística entre os valores calculados (OLIVEIRA et al., 2015).

Com o número de amostras definidas para a área de teste, uma pesquisa de ponto aleatório (centróide do pixel) foi aplicada através da ferramenta vetorial Quantum GIS. Posteriormente, a interpretação visual individual foi verificada (Figura 16), auxiliada pela imagem final da classificação. Os pontos verificados foram classificados em uma matriz de confusão que possibilitou calcular o índice de coeficiencia Kappa.

Figura 16 - Pesquisa de pontos aleatórios. Sistema Quantum GIS.

RESULTADOS E DISCUSSÃO

No sistema minerador RapidMiner, é necessário utilizar diferentes operadores para construir o fluxo da classificação. Os seguintes operadores foram utilizados neste trabalho (Figura 17): ***Retrieve, Set Role e Decision Tree*** (RapidMiner Studio, 2017).

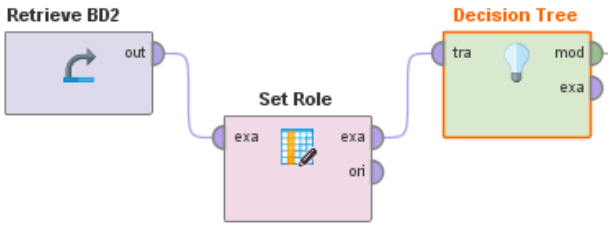


Figura 17 - Operadores do RapidMiner Studio utilizados neste trabalho.

- **Retrieve**: este operador lê os dados a partir de um diretório de armazenamento de dados. Vários formatos

de dados podem ser lidos pelo minerador: csv, xls, bancos de dados SQL, etc. Também é possível ler dados em armazenamento distribuído (na nuvem), pois possui interfaces: Amazon S3, Dropbox, Sales Force, Twitter e Zapier.

- Set Role: este operador é usado para mudar a função de um ou mais atributo. No presente trabalho o atributo class é definido como atributo chave e os demais (Compacity, Angle, Squareness, Circleness, Brightness, Entropy, Maxpixelvalue, Mean, Minpixelvalue, Ratio e Bandmeandiv) como atributo regular.

- Decision Tree: Induz uma árvore de decisão para classificação de dados nominais e numéricos.

A estrutura da árvore de decisão gerada pelo sistema minerador de dados RapidMiner Studio é mostrada na Figura 18. Na Figura (18a) e (18b) são os exemplos de atributo (razão2) e o valor de limiar (> 135.381) da classe de cerâmica escura (18c) que são mostrados no primeiro nó da árvore. Em (18d) representa-se a regra da classe cerâmica escura.

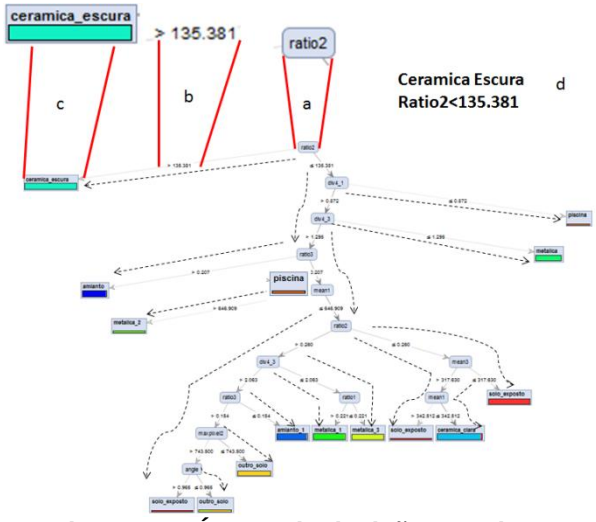


Figura 18 - Árvore de decisão gerada no RapidMiner Studio.

Outros operadores para classificação são disponibilizados no RapidMiner: *Decision Stump, Random Tree* e *Random Forest*. Essas árvores não foram aplicadas neste trabalho, pois cada operador nesse sistema possui especialidade para o manuseio do conjunto de dados (RAPIDMINER, 2017):

- **Operador ID3** - apenas aplicado em dados nominais para a classificação;

- **Operador CHAID** - apenas aplicado em dados nominais para classificação;

- **Operador Decision Stamp** - Pode ser aplicado em dados nominais e numerais, mas gera apenas uma única divisão na árvore, o que não é aplicado a este trabalho;

- **Operador Random Tree** - Este operador não utiliza todos os dados das amostras para construir a árvore de decisão. Ele

utiliza apenas um subconjunto aleatório de atributos para dividir a árvore. Sendo assim, algumas classes podem ficar sem regra e valores de limiares para inserir no InterIMAGE;

- **Operador Random Forest** - Este operador gera um conjunto aleatório de árvores (floresta). As árvores geradas têm a mesma característica da árvore do operador Random Tree, ou seja, é formado por um conjunto aleatório de dados, podendo-se descartar classes no processo de classificação.

Como o RapidMiner, é necessário utilizar diferentes operadores para construir o fluxo de treinamento no minerador de dados KNIME. Os operadores do KNIME utilizados neste trabalho foram os seguintes (Figura 8): Retrieve, Set Role e Decision Tree (RAPIDMINER STUDIO, 2017).

O fluxo e os operadores utilizados do KNIME Analytcs Plataform são apresentados na Figura 19:

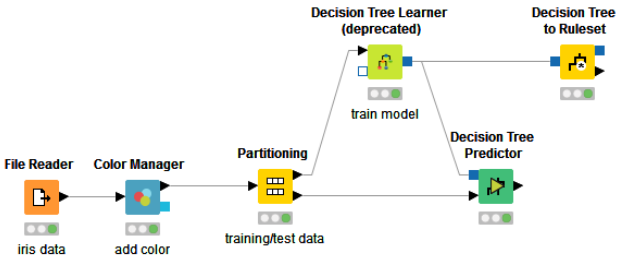


Figura 19 - Fluxo e operadores utilizados no KNIME Analytics Plataform

- **File Reader:** este nó é utilizado para ler dados de um repositório. Pode ser ler vários formatos e extensões. Para este projeto foi lido em extensão CSV;

- Color Manager: define cores para as classes que farão parte da árvore de decisão;

- Partitioning: divide a tabela de entrada em duas partições, permitindo criar: dados de treinamento e dados de teste;

- Decision Tree Learner: este nó induz uma árvore de decisão de classificação na memória principal. Tem como base o algoritmo C45 de Quilan;

- Decision Tree Predictor: este nó induz uma árvore de decisão;

- Decision Tree RoleSet: converte o modelo de árvore de decisão em tabela contendo as regras de forma textual.

Uma porção da árvore de decisão induzida pelo KNIME é apresentada na Figura 20. Em (20a) tem-se o ambiente de navegação (*zoom*). Em (20b), a classe (Cerâmica Clara), que representa o nó e a quantidade de amostras (10) utilizadas para partição. Em (20c) e (20d) representa-se a cor da classe na árvore e o atributo com a regra e valores de limiar

(ratio3>0,1803). O valor entre parênteses em (20b), representado por (9/10), indica que (9) é a quantidade de amostras consideradas confiáveis da classe (telha de amianto) no nó e Y é o número total de amostras usado para este nó.

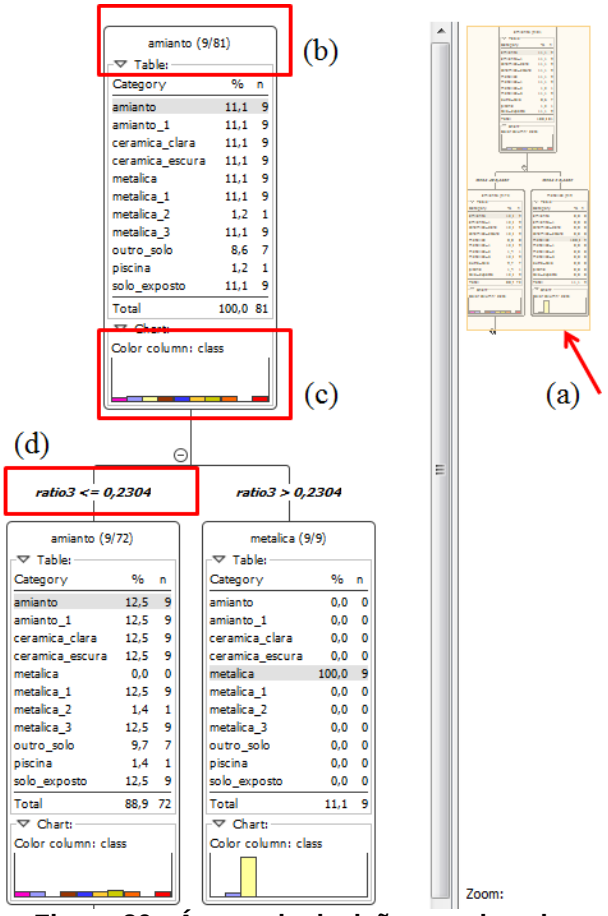


Figura 20 - Árvore de decisão gerada pelo KNIME ANALYTICS PLATFORM

O sistema minerador de dados SIPINA disponibiliza cinco algoritmos de árvore de decisão: ID3, C45, GID3, ASSISTENT 86 e CHAID. A maioria, à exceção do CHAID, é constituída por extensões ao ID3 de Quilan (1986).

Como o RapidMiner e KNIME, é necessário utilizar diferentes operadores para construir o fluxo de treinamento no minerador de dados Orange Canvas. Detalhes do funcionamento e integração do sistema minerador de dados Orange Canvas com o InterIMAGE, utilizando mesma imagem para entrada de dados e configurações, pode ser consultado em Antunes et al. (2016).

O WEKA possui uma biblioteca de algoritmos de aprendizagem de máquina já testada e comparada para classificação do uso do solo, de acordo com os resultados apresentados por Sato et al. (2013). Detalhes do funcionamento e integração do sistema minerador de dados WEKA com o InterIMAGE utilizando a mesma imagem para entrada de dados e configurações neste trabalho pode ser consultado em Antunes et al. (2014).

Não foi possível a inserção de dados gerados pelos mineradores diretamente no InterIMAGE. Todos os limiares gerados nas árvores de decisão pelos diversos mineradores testados neste trabalho, foram inseridos individualmente de forma manual no InterIMAGE, através da interface *TopDown Decision Rule.*

Na Tabela 6 pode ser analizado o desempenho baseado na quantidade de regras da árvore de decisão de cada classificação e o tempo de execução da classificação baseada em objetos no InterIMAGE. É possível verificar na tabela 06 e gráficos 1 e 2 que diferentes mineradores, usando o mesmo algoritmo de classificação apresentam resultados

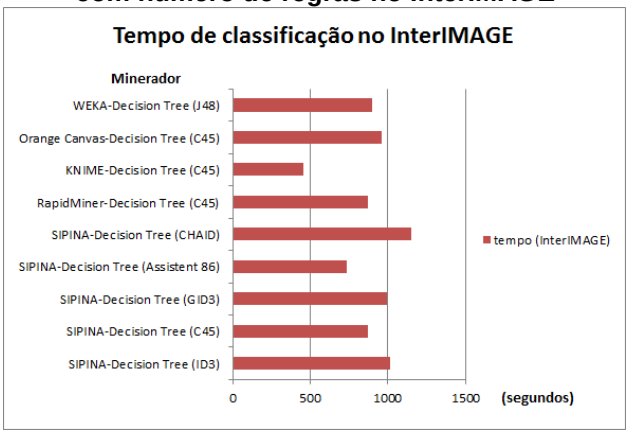
diferentes no que se refere as caracteristicas da árvore de decisão como números de regras.

Table 6 - Diferentes mineradores de dados com diferentes número de regras e tempo da classificação baseada em objeto no InterIMAGE

Sistema	Algoritmo	Regras (und.)	Tempo de classifi (InterIMAGE
SIPINA	Decision Tree (ID3)	16	1018 s
SIPINA	Decision Tree (C45)	14	872 s
SIPINA	Decision Tree (GID3)	16	993 s
SIPINA	Decision Tree Assistent 86	11	735 s
SIPINA	Decision Tree (CHAID)	20	1150 s
RapidMiner	Decision Tree (C45)	14	872 s
KNIME	Decision Tree (C45)	08	455 s
Orange Canvas	Decision Tree (C45)	15	963 s
WEKA	Decision Tree J48	13	900 s

O Gráfico 1 representa a quantidade de regras de cada minerador. A árvore com a estrutura mais compactada foi do KNIME por meio do algoritmo Decision Tree (C4.5). Essa árvore disponibilizou 8 regras a serem inseridas no InterIMAGE. A árvore de maior tamanho (20 regras) foi do sistema SIPINA, por meio do algoritmo CHAID.

Gráfico 1 - Tempo de classificação de acordo com número de regras no InterIMAGE



Na Figura 21 pode-se visualizar o resultado de cada classificação utilizando os diferentes mineradores de dados.

SIPINA-ID3/InterIMAGE	SIPINA-C45/InterIMAGE	SIPINA-GID3/InterIMAGE
SIPINA-Assistent86/InterIMAGE	SIPINA-CHAID/InterIMAGE	KNIME-DT/InterIMAGE
RapidMiner-DT/InterIMAGE	Orange-DT/InterIMAGE	WEKA-J.48/InterIMAGE

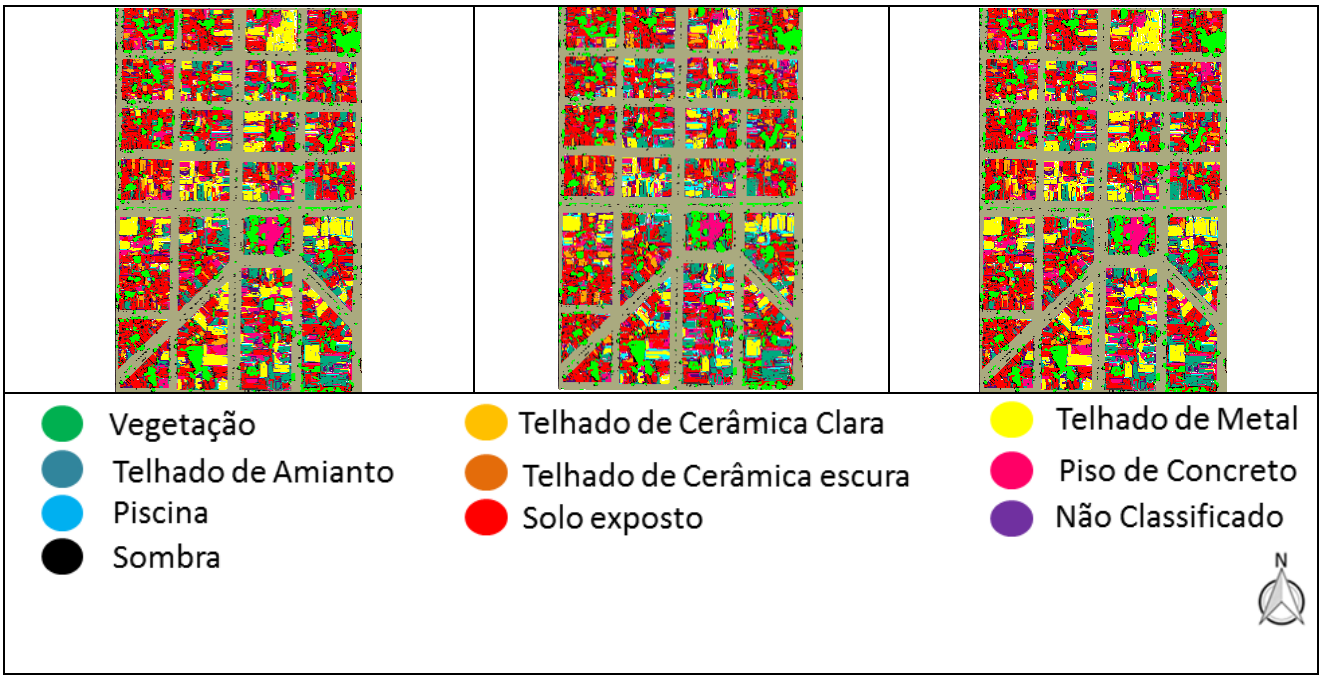


Figura 21 - Resultado da classificação (OBIA) com diferentes mineradores.

Foi aplicada a matriz de confusão para uma análise de concordância da classificação final para cada classificação. A avaliação dos resultados da classificação foi feita baseada no método de concordância KAPPA (BIAS et al., 2014; ANTUNES et al., 2014; PASSO et al., 2013). A Tabela 7 apresenta exemplo da aplicação da Matriz de Confusão com o minerador KNIME.

Table 7 - Matriz de confusão para a classificação KNIME e InterIMAGE

	Amianto	Cerâmica	Metálica	Vias	Solo Exposto	Pav. Concreto	Sombra	Vegetação	Piscina	Não classificado	Total
Amianto	65				6	5				2	78
Cerâmica		8	14		94						116
Metálica	1		63		1	6				3	74
Vias				181							181
Solo Exposto			4		48						52
Pavimento de Concreto	10		4			22				1	37
Sombra	3		3				2			1	9
Vegetação								113			113
Piscina									2	2	4
Não classificado											0
Total	79	8	88	181	149	33	2	113	2	9	-

As maiores confusões apresentadas em todas as matrizes apareceram nas classes de telhado de cerâmica e solo exposto, havendo dificuldades na separação por apresentarem as mesmas características espectrais. Os resultados tiveram a influência de do aumento da proporção das classes Vias e Vegetação na

classificação, no entanto, apresentaram melhor separação e com pouca confusão.

A porcentagem da exatidão total para cada classificação por meio do índice de exatidão Kappa e o resultado do teste de significância (teste z) é apresentado na Tabela 8 e Gráfico 2.

Table 8 - Resultado do índice Kappa e o resultado do teste z para cada classificação do InterIMAGE por meio da integração com os mineradores: SIPINA, RapidMiner Studio, Knime Analytics Platform, Orange Canvas e WEKA.

	SIPINA ID3	SIPINA C45	SIPINA GID3	SIPINA Assistent86	SIPINA CHAID	RAPIDMINER C45	KNIME C45	ORANGE C45	WEKA J48
EXATIDÃO GLOBAL	70%	72%	73%	73%	73%	79%	75%	83%	81%
KAPPA	0,66	0,68	0,70	0,70	0,70	0,77	0,73	0,81	0,78
VARIANCIA KAPPA	0,000406	0,000354	0,000382	0,000391	0,000387	0,000344	0,000387	0,000268	0,00032
SIPINA ID3 (16 regras)	calc z 32,25	0,72	1,42	1,42	1,42	4,02	2,48	4,62	4,44
SIPINA C45 (14 regras)	0,72	calc z 35,56	0,74	0,73	0,73	3,40	1,84	5,20	3,84
SIPINA GID3 (16 regras)	1,42	0,74	calc z 35,29	0,00	0,00	2,60	1,08	4,31	3,01
SIPINA Assistent 86 (11 regras)	1,42	0,73	0,00	calc z 34,85	0,00	2,58	3,00	4,28	2,99
SIPINA CHAID (20 regras)	1,42	0,73	0,00	0,00	calc z 35,05	2,59	1,08	4,29	3,00
RAPIDMINER C45 (14 regras)	4,02	3,40	2,60	2,58	2,59	calc z 40,97	1,48	1,62	0,39
KNIME C45 (8 regras)	2,48	1,84	1,08	1,07	1,08	1,48	calc z 40,05	3,12	1,87
ORANGE C45 (15 regras)	5,77	5,20	4,31	4,28	4,29	1,62	3,12	calc z 48,78	1,23
WEKA J48 (13 regras)	4,44	3,84	3,01	2,99	3,00	0,39	2,05	1,23	calc z 42,80

O teste Z, o qual teve como base os valores e a variância do índice Kappa, permitiu a comparação entre os resultados das diferentes classificações com o InterIMAGE e mineradores de dados. Os valores foram organizados na Tabela 4 cuja diagonal principal equivale ao resultado do Teste Z para um único índice Kappa e os demais valores mostram o resultado da correlação entre dois índices.

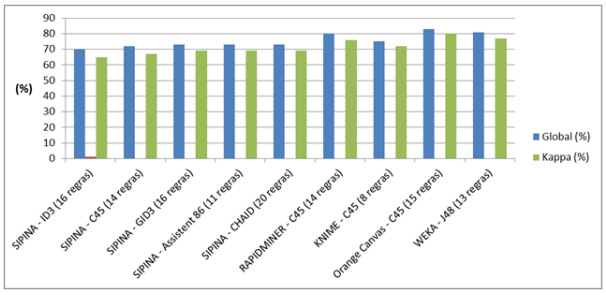
Ao analisar a tabela 4, observa-se que as classificações que resultaram em índice 0,0 (zero), não possuem significância, pois possuem o mesmo índice Kappa, causando assim uma inexistência de dados para o Teste Z. É o caso, por exemplo, da classificação InterIMAGE/SIPINA (GID3) e InterIMAGE/SIPINA (CHAD), ambos com índices 73% Kappa. Como estes resultados de classificação não possuem um resultado de significância, não faz sentido à comparação entre eles e a correlação não foi considerada nos cálculos.

O valor Z foi “tabelado” ao nível de 5% de probabilidade, igual a 1,96. Levando em consideração o valor Z calculado conforme apresentado na Tabela 4 e maior que 1,96, consideramos que o resultado e diferença entre as classificações foram significativas, concluindo-se que o resultado das classificações é estatisticamente diferente, é o caso dos valores apresentado de cor vermelha na Tabela 4.

A maior significância dos resultados foi entre Orange Canvas/SIPINA (ID3) que atingiu um valor de 4,62 e WEKA/SIPINA (ID3) que atingiu um valor de 4,44. As classificações do InterIMAGE utilizando os algoritmos do SIPINA (ID3, C45, GID3, Assistent 86 e CHAID) tiveram valores de significância abaixo de 1,96 , demostrando que não houve diferença estatística.

A quantidade de regras inseridas no InterIMAGE não interferiu no resultado da classificação baseada em objetos. Uma análise (Gráfico 2) pode ser feita por meio do resultado da integração KNIME/InterIMAGE com 8 regras, que atingiu acurácia de 73% para o índice Kappa e o da integração SIPINA(CHAID)/InterIMAGE, com 20 regras, com acurácia 70% para o índice Kappa, ou seja, a integração SIPINA(CHAID)/InterIMAGE utilizando maior quantidade de regras (20 regras) não obteve melhor resultado que KNIME/InterIMAGE com menor quantidade de regras (8 regras).

Gráfico 2 - Resultado da classificação (Kappa) do InterIMAGE por meio da integração com os mineradores: SIPINA, RapidMiner Studio, KNIME Analytics Platform, Orange Canvas e WEKA



CONCLUSÃO

Os resultados apresentados mostraram a possibilidade de trabalhar com outros mineradores de dados, além do WEKA, para um processo de classificação baseada em objetos, por meio do sistema InterIMAGE. Outros mineradores de código aberto, além do WEKA, utilizados neste trabalho foram: SIPINA, RapidMiner Studio, KNIME Analytics Platform e Orange Canvas.

Todos os sistemas mineradores de dados testados nesta pesquisa foram fáceis de operar, mesmo com interfaces e visualizações diferentes.

O SIPINA não teve um resultado muito bom utilizando seus algoritmos para este trabalho. O melhor índice de acurácia atingido foi 70% Kappa (GID3, Assistent 86 e CHAID) e os resultados das classificações OBIA com seus algoritmos não teve diferença estatística entre eles, de acordo com o teste Z.

Orange Canvas (81% Kappa) e WEKA (78% Kappa) apresentaram os melhores resultados de classificação e índices de significância entre eles foram menor 1,96, ou seja, não houve diferença estatística, possibilitando ao analista OBIA escolher o mais interessante para seu projeto.

Independente de qual sistema minerador de dados ou algoritmo utilizado, a quantidade de regras influencia no tempo de processamento do InterIMAGE mas não influencia no resultado (acurácia) da classificação baseada em objetos. Sistemas mineradores diferentes, utilizando o mesmo algoritmo (C.45), podem apresentar uma estrutura de árvore diferente devido a vários fatores como características internas dos

softwares de mineração de dados, diferentes amostragens (com diferentes efeitos sobre a acurácia) e outras.

AGRADECIMENTOS

Os autores reconhecem o apoio prestado pela CAPES (Coordenação de Aperfeiçoamento de Pessoal de Nível Superior).

REFERÊNCIAS BIBLIOGRÁFICAS

ADDINK, E. A.; DE JONG, S. M.; PEBESMA, E. J. The importance of scale in object-based mapping of vegetation parameters with hyperspectral imagery. Photogrammetric Engineering and Remote Sensing, v. 73, n. 8, 2007, p. 905-912.

AMORIM, Rômulo Magalhães; ALBARICI, Fabio Luiz; DEL PINO, Miguel Angel Isaac Toledo. Mapa digital do escoamento superficial por meio de imagens de sensor remoto na sub-bacia do Rio Moji Guaçu-MG. Revista Brasileira de Geografia Física, v. 9, n. 3, p. 881-896, 2016.

ANTUNES, R. R. et al. Desenvolvimento de técnica para monitoramento do cadastro urbano baseado na classificação orientada a objetos. Estudo de caso: Município de Goianésia, Goiás. Revista Brasileira de Cartografia, v. 67, n. 2, p. 357-372. Brasília, 2015.

ANTUNES, R. R. et al. Integration of open-source tools for object-based monitoring of urban targets. In: GEOBIA 2016: Solutions and synergies. University of Twente Faculty of Geo-Information and Earth Observation (ITC). 2016.

BAATZ, M.; SCHAPE, A. Object-oriented and multi-scale image analysis in semantic networks. The 2nd International Symposium: Operationalization of Remote Sensing, New Methodologies, 16-20 August 1999.

BIAS, E. et al. Application of imagery analysis based on objects as a tool for monitoring the urban cadaster in small municipalities. South Eastern European Journal of Earth Observation and Geomatics. Greece: Aristotle University of Thessaloniki, 2014.

BLASCHKE, T. What's wrong with pixels? Some recent developments interfacing remote sensing and GIS. GeoBIT/GIS, v. 6, 2001, p. 12-17.

BLASCHKE, T. Object-based contextual image classification built on image segmentation. Advances in techniques for analysis of remotely sensed data. IEEE Workshop on, 2003, p. 113-119.

BLASCHKE, T. Object-based image analysis: a new paradigm remote sensing? ASPRS, Anual Conference. Baltimore, Maryland, 2013.

BLASCHKE, T.; BURNETT, C.; PEKKARINEN, A. Image segmentation methods for object-based analysis and classification. In: Remote sensing image analysis: Including the spatial domain. Netherlands, 2004, p. 211-236.

BOLFE, Édson Luis et al. Avaliação da classificação digital de povoamentos florestais em imagens de satélite através de índices de acurácia. Revista Árvore, Brazilian Journal of Forest Science, v. 28, n. 1, 2004.

BRACHMAN, R. J. What IS-A Is and isn't: an analysis of taxonomic links in semantic networks. Computer, v. 16, p. 30-36, October 1983.

BREIMAN, L. et al. Classification and regression trees. CRC press, 1984.

CADENA, G. T. Classificação dos tipos de pavimentos das vias urbanas a partir de imagem de alta resolução espacial por meio de análise orientada a objeto. [Dissertação de mestrado em Ciências Cartográficas, Universidade Estadual Paulista, p. 114]. Presidente Prudente, SP, 2011.

CESTNIK, B.; KONONENKO, I.; BRATKO, I. ASSISTANT 86: A knowledge elicitation tool for sophistical users, Proc. of the 2nd European Working Session on Learning, 1987, p. 31-45.

COHEN, J. A. Coefficient of agreement for nominal scales. Educational and measurement, v. XX, n. 1, 1960, p. 37-46.

CONGALTON, R. G.; GREEN, K. Assessing the accuracy of remotely sensed data: principles and practices. Boca Raton-USA: Lewis Publisher, 1999.

COSTA, G. A. O. P. et al. Knowledge-based interpretation of remote sensing data with the InterIMAGE System: major characteristics and recent developments. GEOBIA 2010. Gent, Belgium, 2010.

DE GRANDE, Thallita Oliveira; DE ALMEIDA, Tati; CICERELLI, Rejane Ennes. Classificação orientada a objeto em associação às ferramentas reflectância acumulada e mineração de dados. Pesquisa Agropecuária Brasileira, v. 51, n. 12, 2017, p. 1983-1991.

DEMSAR, J.; CURK, T.; ERJAYEC, A. Orange: data mining toolbox in Python; Journal of Machine Learning Research, Aug/2013, p. 2349-2353.

DORREN, L. K. A.; MAIER, B.; SEIJMONSBERGEN, A. C. Improved landsat-based forest mapping in steep mountainous terrain using object-based classification. Forest Ecology and Management, v. 183. Elsevier, Amsterdam, 2003, p. 31-46.

DOS ANJOS et al. Análise do nível de legenda de classificação de áreas urbanas empregando imagens multiespectrais e hiperespectrais com os métodos árvore de decisão c4.5 e floresta randômica. Boletim de Ciências Geodésicas, v. 23, n. 2, 2017, Curitiba, Brasil.

FAULHABER, A. AI Tools: A short introduction to Yale. department of mathematics and computer science. Saarbrücken, Germany: Saarland University, 2007.

FAYYAD, U. Branching on attribute values in decision tree generation. AI Group, M/S 5253660, Jet Propulsion Laboratory, California Institute of Technology. Pasadena, CA. 1994.

FERREIRA, Rodrigo da Silva. InterIMAGE cloud platform: the architecture of a distributed platform for automatic, object-based image interpretation. [Tese de doutorado em engenharia elétrica pela Pontifícia Universidade Católica do Rio de Janeiro – PUC-Rio]. Rio de Janeiro, 2015.

FERREIRA, R.S. et al. A set of methods to support object-based distributed analysis of large volumes of earth observation data. IEEE Journal of Selected Topics in Applied Earth Observations and Remote Sensing (JSTARS), v. 10, n. 2, 2017, p. 681-690.

GHOLAP, Jay. Performance tuning of j48 algorithm for prediction of soil fertility. Pune, Maharashtra, India: Dept. of Computer Engineering. College of Engineering, 2012.

GRIPPA, T. et al. An open-source semi-automated processing chain for urban OBIA classification. In: GEOBIA 2016: Solutions and Synergies. 14 September 2016. University of Twente Faculty of Geo-Information and Earth Observation (ITC).

HELLDEN, U.; STERN, M. Evaluation of landsat imagery and digital data for monitoring desertification indicators in Tunisia. Proc. 14th. Int. Symp. on Rem. Sens. of Environ., p. 1601-1611, 1980.

HSSINA, B. et al. A comparative study of decision tree ID3 and C4.5. (IJACSA) International Journal of Advanced Computer Science and Applications, Special Issue on Advances in Vehicular Ad Hoc Networking and Applications. TIAD laboratory, Computer Sciences Department, Faculty of sciences and techniques. Sultan Moulay Slimane University, Morocco, 2014.

INTERIMAGE WIKI. Attributes description. Pontífera Universidade Católica do Rio de Janeiro (PUC-Rio) e Instituto Nacional de Pesquisa Espacial (INPE). 2014. Disponível em: <http://wiki.dpi.inpe.br/doku.php?id=interimage:attributes_description> Acesso em: 16 jun. 2017.

INTERIMAGE. Manual do usuário. 2010. Disponível em: <http://www.lvc.ele.puc-rio.br/projects/interimage/pt-br/documentacao/> Acesso em: 09 abr. 2016.

KASS, G. An exploratory technique for investigating large quantities of categorical data. Applied Statistics, v. 29, n. 2, 1980, p. 119-127.

KAUR, A.; SINGH, S. Classification and selection of best saving service for potential investors using decision tree – Data mining algorithms. International Journal of Engineering and Advanced Technology (IJEAT), v. 2, n. 4, April 2013.

KLECKA, W. R. Discriminant analysis. Beverly Hills, California: SAGE Publications, 1980, p. 71.

KNIME ANALYTICS PLATFORM. Open for innovation. 2017. Switzerland. Disponível em: <https://www.knime.org/> Acesso em: 09 jun. 2017.

MAVRANTZA, O.; ARGIALAS, D. An object-oriented image analysis approach for the identification of geologic lineaments in a sedimentary geotectonic environment. In:

BLASCHKE, T.; LANG, S.; HAY, G. J. Object-based image analysis. Berlin, German. Springer, 2008.

MEINEL, G.; NEUBERT, M. A comparison of segmentation programs for high resolution remote sensing data. Commission VI in Proceeding of XXth ISPRS Congress. International Society for Photogrammetry and Remote Sensing. Istanbul, Turkey, 2004.

MENESES, Paulo Roberto; SANO, Edson. Classificação pixel a pixel de imagens. In: MENESES, Paulo Roberto; ALMEIDA, Tati. Introdução ao processamento de imagens de sensoriamento remoto. 1. ed., v. 1. Brasília: CNPq, 2012.

MORGAN, J. N.; SONQUIST, J. A. Problems in the analysis of survey data, and a proposal. Journal of the American Statistical Association, n. 58, 1963, p. 415-434.

OLIVEIRA, F. P. D. et al. Mapeamento de florestas monodominadas por Myracrodruon urundeuva com imagens TM-Landsat 5 e Rapideye. Floresta e Ambiente, v. 22, n. 3, p. 322-333. Instituto de Florestas da Universidade Federal Rural do Rio de Janeiro, 2015.

ORLANDO, P.; LA ROSA, E. Object oriented methodology for change detection technique: the case of Scopello-Silicy. South-Eastern European Journal of Earth Observation and Geomatics. v. 3, n. 2S, Greece: Aristotle University of Thessaloniki, 2014.

OTUKEI, J. R.; BLASCHKE, T. Land cover change assessment using decision trees, support vector machines and maximum likelihood classification algorithms. International Journal of Applied Earth Observation and Geoinformation. Elsevier, 2010.

PASSO, D. P. et al. Uso do sistema InterIMAGE para a identificação de alvos urbanos em imagens do satélite Worldview II. Revista Brasileira de Cartografia, v. 6, 2013.

QGIS BRASIL. Comunidade de usuários QGIS Brasil. 2015. Disponível em: <http://qgisbrasil.org/> Acesso em: 14 abr. 2017.

QUILAN, J. Induction of decision trees, v. 1. Boston (USA): Machine Learning, Kluwer Academic Publishers, 1986, p. 81-106.

RAKOTOMALALA, R. Introduction of a decision tree using SIPINA. Tutorial. Departamento de Informática e Estatística. University Lyon, France. 2008.

RAPIDMINER STUDIO. Manual. Boston, USA. 2014. Disponível em: <http://docs.RapidMiner.com/downloads/RapidMiner-v6-user-manual.pdf> Acesso em: 09 jul. 2017.

RICH, E. Inteligência artificial. São Paulo: McGraw-Hill, 1988.

SATO, L. Y. et al. Análise comparativa de algoritmos de árvore de decisão do sistema WEKA para classificação do uso e cobertura da terra. XVI Simpósio Brasileiro de Sensoriamento Remoto, 2013, p. 2353-2360.

TAN, P. N.; STEINBACH, M.; KUMAR, V. Introduction to data mining. Boston, MA, USA: Addison-Wesley Longman Publishing, 2005.

WILGES, B. et al. Avaliação da aprendizagem por meio de lógica de fuzzy validado por uma Árvore de Decisão ID3. Novas Tecnologias na Educação, v. 8, n. 3. Centro interdisciplinar de novas tecnologias na educação – CINTED. Universidade Federal do Rio Grande do Sul – UFRGS. Dez./2010.

WITTEN, I. H. et al. Weka: Practical machine learning tools and techniques with Java implementations. Department of Computer Science, University of Waikato, New Zealand, 1999.

ZANETTI, J.; BRAGA, F.L.S.;DUARTE, D.C.O. Comparação dos métodos de classificação supervisionada de imagem máxima verossimilhança, distância euclidiana, paralelepípedo e redes neurais em imagens vant, utilizando o método de exatidão global, índice kappa e o tau. IV Simpósio Brasileiro de Geomática – SBG2017. II Jornadas Lusófonas - Ciências e Tecnologias de Informação Geográfica - CTIG2017. Presidente Prudente – SP, Brasil, 2017.

4 - ARTIGO 3

PROOF OF CONCEPT OF A NOVEL CLOUD COMPUTING APPROACH FOR OBJECT-BASED REMOTE SENSING DATA ANALYSIS AND CLASSIFICATION

R. R. Antunes[a], T. Blaschke[c], D. Tiede[c], E. S. Bias[a], G. A. O. P. Costa[b], P. Happ[d]

[a]Federal University of Brasilia, Brazil – rodrigorantunes@hotmail.com, edbias@unb.br
[b]University of Salzburg, Austria – Thomas.Blaschke@sbg.ac.at, dirk.tiede@sbg.ac.at
[c]Rio de Janeiro State University, Brazil – gilson.costa@ime.uerj.br
[d]Pontifical Catholic University of Rio de Janeiro, Brazil – patrick@ele.puc-rio.br

ABSTRACT

Advances in the development of Earth observation data acquisition systems have led to the continuously growing production of remote sensing datasets, for which timely analysis has become a major challenge. In this context, distributed computing technology can provide support for efficiently handling large amounts of data. Moreover, the use of distributed computing techniques, once restricted by the availability of physical computer clusters, is currently widespread due to the increasing offer of cloud computing infrastructure services. In this work, we introduce a cloud computing approach for object-based image analysis and classification of arbitrarily large remote sensing datasets. The approach enables exploiting machine learning methods in the creation of classification models, through the use of a web-based notebook system. A prototype of the proposed approach was implemented with the methods available in the InterCloud system integrated with the Apache Zeppelin notebook system, for collaborative data analysis and visualization. In this implementation, the Apache Zeppelin system provided the means for using the scikit-learn Python machine learning library in the creation of a classification model. In this work we also evaluated the approach with an object-based image land-cover classification of a GeoEye-1 scene, using resources from a commercial cloud computing infrastructure service provided. The obtained results showed the effectiveness of the approach in efficiently handling a large data volume in a scalable way, in terms of the number of allocated computing resources.

INTRODUCTION

During the course of the past few decades, outstanding advances were achieved in the development of Earth observation (EO) remote sensing (RS) data acquisition systems. Developments in sensor technology were responsible for a continuous increase in spatial and spectral resolution of optical sensors, and similar progress was seen in radar- and laser-based imaging. The number and variety of EO platforms have also grown significantly, considering especially the dissemination of unmanned aerial vehicles and the development of small-scale orbital platforms.

The direct result of such technological advances is the daily production of massive amounts of RS data. The NASA EOSDIS project, for instance, produces about 12 TB of data daily [1]. This scenario leads to new challenges, related to the capacity of handling huge volumes of data with respect to computational techniques and resources [2]. In this sense, remote sensing data handling may be considered a big data problem, due to the high data volume, variety, and generation velocity [3, 4].

In this context, distributed computing technology can provide valuable support for efficiently handling large RS datasets, as data can be partitioned into smaller subsets, which are processed in parallel by different computing units. Moreover, the use of distributed computing techniques, once restricted by the availability and access to physical computer clusters, is currently widespread, as the offer of cloud computing infrastructure services at affordable costs has become commonplace [5].

Cloud computing delivers powerful, scalable infrastructure for the processing of large-scale datasets. In a cloud environment, a set of virtual computing components is delivered on demand, offering data access transparency and elastic provisioning of fail-safe resources in a pay-as-you-go service model [6].

Furthermore, many works in the past two decades have exposed both the limitations of pixel-based techniques in the analysis of high spatial resolution RS image data, and the advantages of object-based image analysis (GEOBIA) [7, 8]. Rather than dealing with individual pixels, object-based approaches aim at classifying image segments through the analysis of their attributes, which may include spectral, textural or morphological characteristics and topological relationships among segments.

The capacity of dealing with georeferenced raster and vector data is essential to GEOBIA approaches. But handling these types of data in a distributed

environment is not a trivial matter [9]. The major problem has to do with distributing segments and image data while keeping track of their spatial relationships, and, at the same time, minimizing communication among processing nodes when performing spatially-aware operations.

Recently, Ferreira et al. [6] proposed and evaluated a set of methods that support distributed object-based operations, such as the computation of spectral, morphological and topological properties, and knowledge-driven classification. Some of those methods were employed in the construction of a tile-based distributed image segmentation method, introduced by Happ et al. [10], and of a machine learning distributed classification framework [11]. The methods proposed in [6] were implemented using the MapReduce programming model [12] in a system called InterCloud.

In this work, we build upon the InterCloud framework, and investigate its integration with a cloud-based notebook interface system, the Apache Zeppelin [13], which supports collaborative data analytics and visualization, and provides for the interpretation of data processing commands and workflows over distributed data.

Furthermore, we implemented an object-based classification scheme that relies on a machine learning software library, the scikit-learn Python library [14], and evaluated the approach on an urban land-cover object-based image interpretation application, executed completely in a cloud computing infrastructure environment.
The main contributions of this work are:

(1) We introduce a cloud computing approach for object-based image analysis and classification of arbitrarily large remote sensing data sets.
(2) We propose a way of exploiting machine learning methods in the definition of classification models for remote sensing data, through the use of a collaborative, web-based notebook system.
(3) We present the evaluation of an implementation of the proposed approach on a land-cover classification application, executed over cloud computing infrastructure, considering its potential scalability.

The remainder of this paper is organized as follows. Related work regarding distributed georeferenced spatial data processing is presented in Section II. The

components and processing steps of the proposed approach are presented in Section III. The design of the experiments devised to evaluate an implementation of the proposed approach are presented in Section IV. The experimental results are presented and discussed in Section V. Section VI presents conclusions and directions for future research.

RELATED WORK

Different distributed solutions have recently been proposed aiming at efficiently handling large volumes of RS data. The work of Golpayegani and Halem [15] implements distributed image analysis operations but focuses only on raster data. The work of Chen et al. [16] focuses on coupling the OpenGIS Web Processing Service (WPS) with a distributed cloud-based processing environment; the work is, however, restricted to pixel-based analysis.

The work of Cappelaere et al. [17] implements a full spectral unmixing chain in the Web Coverage Processing Service (WCPS), a cloud-based image processing framework. Tan et al. [18] developed an agent-based geospatial service chain. There are other studies that focus on RS image analysis [19, 20, 21], but they also only support pixel-based analyses.

Relatively few studies have proposed techniques for storing and querying large geo-information datasets [22, 23, 24]. Systems like Hadoop-GIS [25] and SpatialHadoop [26] offer complete spatial data storage and spatial query execution, but they do not handle image data.

Google recently deployed a web-based service, the Google Earth Engine (GEE) [27], which is capable of integrating raster and vector RS data processing. GEE's underlying framework and methods are, unfortunately, not disclosed, thus hindering scientific investigation.

Regarding machine learning, few works focusing on handling big EO datasets can be found in the literature [28], but they are more concerned with the training phase than with the classification procedure itself. Quirita et al. [11] propose a distributed architecture for supervised classification of large EO datasets, which supports the integration of machine learning methods. In that work, the proposed architecture is employed in pixel-based, hyperspectral image analysis.

To the best of our knowledge, the work of Ferreira et al. [6], which introduces the methods implemented in InterCloud, is the only available open solution that supports distributed object-based RS image analysis in the cloud. Image segmentation in the application reported in [6] was performed off-line, i.e., not in a distributed fashion.

Lastly, a literature search did not reveal any work that supports object-based, distributed RS image analysis, integrated with a web-based notebook infrastructure, such as Apache Zeppelin, for data mining and classification modeling.

METHOD

The proposed methodology consists of five main steps, organized in the processing chain depicted in Figura 22: (a) segmentation; (b) computation of features; (c) sample selection; (d) classification modeling; and (e) classification.

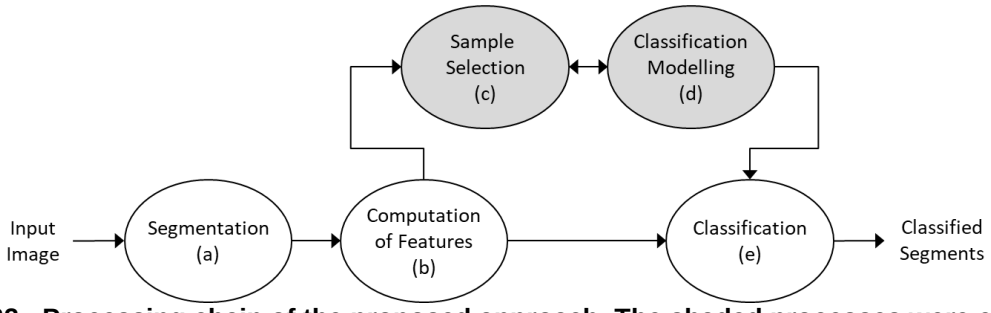


Figura 22 - Processing chain of the proposed approach. The shaded processes were executed off-line.

After distributed segmentation, a number of feature values of the generated segments are computed in a distributed way. Then, in an off-line procedure, representative samples of the classes of interest are selected, and this reference set is subsequently used in the generation of a classification model, also off-line. The final step is the distributed classification of the complete set of segments.

In this work, processes (a), (b) and (e) were carried out with InterCloud. The sample selection procedure (c), was performed manually, with the aid of the QuantumGIS system. The classification modeling procedure was carried out with the aid of a Zeppelin notebook, and the scikit-learn Python library.

Note that after the classification model has been generated, the classification process can be carried out in a batch mode, using only procedures (a), (b) and (e).

This could be the case if the object-based classification process, counting on the same classification model, would be executed for another geographic area, covered by a RS image with similar characteristics, i.e., acquired by the same sensor.

In the next sections, we detail the processing chain steps.

A. Distributed Segmentation

The distributed segmentation procedure was performed using the method described in Happ et al. [10], which is implemented in the InterCloud system. The method comprises a strategy for performing region-growing segmentation in a distributed environment. The strategy tackles the distributed segmentation by (a) splitting an input image into tiles, in order to generate independent datasets for distributed computing; (b) performing region-growing segmentation independently for each tile; and (c) efficiently stitching neighboring segments that touch the tile borders, thus eliminating the artifacts, i.e., straight segment outline sections over tile borders, resulting from the independent distributed segmentations. Moreover, the distributed region-growing strategy is scalable and capable of efficiently handling very large RS images.

The method offers three alternatives for stitching segments that touch tile borders: Simple Post-Processing (SPP); Hierarchical Post-Processing (HPP); and Hierarchical Post-Processing with Re-segmentation (HPPR). The latter alternative is more time consuming than the others, but it is able to deliver artifact-free results, which is the reason why it was selected for the implementation of the proposed approach.

The HPPR involves iterative processing through the levels of the quad-tree formed by a hierarchy of geo-cells, a geographical grid based on a given coordinate reference system, which defines the dimensions of the image tiles. The HPPR solution relies on iterative steps to ensure that all bordering segments are accounted for. Basically, it groups all segments that touch the borders of geo-cells at one iteration, then it discards those segments and generates new segments by performing the region-growing procedure over the pixels once covered by the discarded segments. This procedure is performed iteratively until the coarser geo-cell level is reached, i.e., the one in which the input image is covered by a single geo-cell. Figura 23 shows examples of segments generated in an iteration of the method.

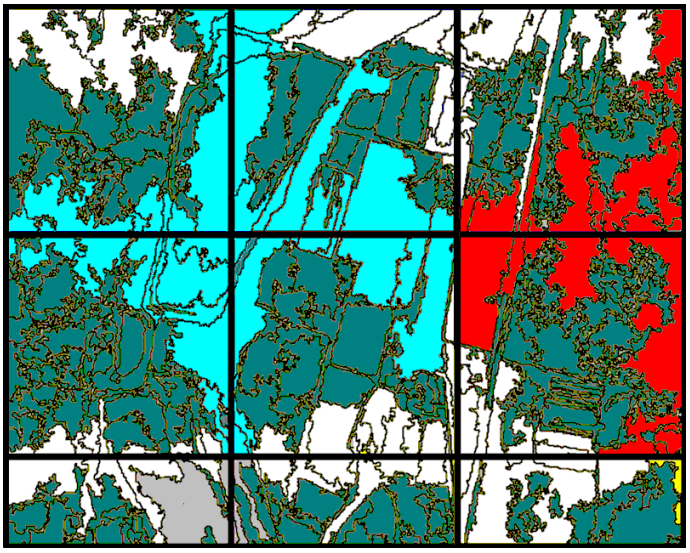


Figura 23 - Segments generated in an iteration of the method (HPPR). All segments that touch the borders of geo-cells are discarded and the corresponding image regions are re-segmented.

The particular region-growing segmentation algorithm implemented in InterCloud's distributed segmentation module in based on the multiresolution region-growing algorithm proposed in Happ et al. [10], which considers every pixel as a region-growing seed and supports the use of spectral and morphological homogeneity criteria in the definition of the segment merging rule.

B. Computation of Features

The computation of feature (or property) values of the segments generated in the previous step was also implemented with InterCloud [6]. The main design concern in the set of methods devised for the system was the capacity of distributing the processing of raster (i.e., image tiles) and vector (i.e., image segments) data in such a way that the communication overhead among distributed processes is as small as possible [9].

InterCloud's underlying methods [6] rely on a specific spatial indexing mechanism, which is based on a geographic grid, so-called geo-cell grid, defined over a selected coordinate reference system, as mentioned in the previous section. Besides determining the division of the input image data into image tiles, the geo-cells are also used to index the image segments they overlap. If a segment overlaps more than one geo-cell, it may be replicated for each overlapping geo-cell and

indexed accordingly. There are also different geo-cell levels that are used to group image segments for particular distributed operations.

A set of distribution strategies define different ways to process (i.e., group, retrieve, and replicate) image tiles and segments for different types of operations. Four distribution strategies were proposed in [6] to support distribution of object-based operations: spatial-blind; spatial aware with replication; spatial aware without replication; and recursive. The recursive strategy supports image segmentation, as described in the previous section. The other three strategies support, among other capabilities, the computation of segment features.

Spatial-blind feature computation operations include the calculation of morphological features, such as area, compactness, asymmetry, rectangular fit, and so on. Such operations do not need to consider the spatial locality of the input segments, which are processed independently.

The spatial aware with replication strategy supports operations that rely on the spatial location of data, such as the calculation of spectral and topological features. Segments are replicated for all the geo-cells they intersect and then grouped by geo-cell. For each segment in a group, the computation is performed considering only the intersection between the segment and the corresponding image tile. Those partial results are later combined to produce the operations' final results.

The spatial aware without replication strategy is used in multiscale analysis operations. It also relies on spatial locality, but segments are not replicated. Initially, segments are grouped by their parent segment ID rather than by geo-cells. Hierarchical features can be computed once a parent segment and its child segments are gathered on the same cluster node. Aggregated features can be computed based on any feature child segments have in common.

C. Sample Selection

To support the classification modeling step of the proposed approach, an appropriate number of samples of each class of interest needs to be selected. In this work, the selection of segment samples representative of each class was performed visually, by overlaying the input image with the borders of the segments produced in the segmentation step (Section III.A). The procedure was performed with the aid of the QuantumGIS software.

The sample selection step was carried out in an interactive fashion, regarding the classification step. This means that candidate samples were evaluated by taking into consideration the whole set of samples, in the attempt of determining a set of samples for each class that best represent its within-class variability, and detecting outliers. In this work, the procedure generated 30 to 60 samples per class.

D. Classification Modeling

The classification modeling step is responsible for the analysis of the feature space, and the creation of a classification model for one or more classes of interest.
In this work, we propose the use of a web-based notebook system, in this case, the Apache Zeppelin for providing data visualization tools and an interface layer to different systems for data exploration and classification. Zeppelin was devised as a tool for data scientists to collaborate on large-scale data exploration and visualization projects. In a practical sense, data processed by the integrated interpreter systems can be analyzed visually with the aid of customized graphs and charts, organized in a virtual notebook.

Zeppelin is currently deployed with pre-created interfaces for many distributed data processing systems and programing languages, such as Beam, Cassandra, ElasticSearch, Flink, Groovy, Pig, PostgreSQL, Python, R, and many more [13]. Additionally, new interpreters can be attached to the notebook interface through its application programming interface (API).

In this work, we used a powerful machine learning Python library, scikit-learn, for the creation of a classification model. The scikit-learn library is considered one of the most popular machine learning libraries among all programing languages, as it is a collaborative, open-source initiative, and contains a large number of methods for data mining and data analysis [14].

So, after making the scikit-learn library available in the cloud environment (i.e., installing it in the cluster nodes dedicated for this task), we are able to execute the methods in the library over the distributed data (in this case, segment samples), by simply stating the respective Python commands in the Zeppelin notebook graphical user interface (GUI).

In this work, the decision tree classification method available in scikit-learn was used for classification modeling. A decision tree was inducted through the algorithm, and the corresponding classification rules (based on thresholding feature values) compose the classification model.

E. Classification

After the classification model was defined in the previous step, it was translated into classification decision rules in InterCloud, so that the actual classification of all image segments can be carried out.

One classification rule, combining all sharp decisions based on the feature values thresholds defined in the classification tree, was created for each class, in the form of Pig Latin [29] scripts, which are interpreted by InterCloud's engine during distributed classification execution, which is a spatial-blind operation in terms of what was stated in Section III.B.

EXPERIMENTAL SETUP

To validate and evaluate the proposed approach, experiments were performed using a virtual cluster in a commercial cloud computing infrastructure. In the experiments, a land-cover classification application of a very high-resolution optical satellite image was carried out, with various computer cluster configurations.

A. Cloud Environment

The experiments were conducted over cloud computing infrastructure provided by Amazon Web Services (AWS). Amazon Simple Storage Service (S3) was used to store the input and output data, as well as the required programs and libraries. Amazon Elastic MapReduce (EMR) was used to dynamically build and manage clusters of Amazon Elastic Compute Cloud (EC2) instances. EC2 instances were of type m3.large virtual machines, having Intel Xeon E-5-2670 v2 processors operating at 2.5GHz with a 64-bit architecture, 15 GB of RAM and 2 disks with 40GB using SSD technology. Each processor has 4 physical cores and 8 logical cores. All machines have Apache Hadoop 2.73 and Apache Pig 0.14.0 installed.

In the experiments, the batch classification procedure, which encompasses the segmentation (a); features computation (b); and classification (e) steps of the proposed approach was executed in computer clusters of different sizes: with 2, 4, 8, 16 and 32 virtual machines. It is important to mention that one cluster node is reserved for the Hadoop JobTracker (the master node), which is responsible for scheduling and managing the tasks and is not available for performing other processing tasks.

For the classification modeling (d) step, two similar, additional virtual machines were used to run the Apache Zeppelin notebook system and the scikit-learn Python library functions. The following Python modules were installed on those machines: NumPy 1.8.2; SciPy 0.13.3; and scikit-learn 0.19.1 (which requires the first two modules).

B. Test Site and Image Data

The test site is in the Goianésia municipality, in the state of Goiás, Central-East Region of Brazil. The site contains dense to sparse urban areas, surrounded by farmland with small patches of natural (forested) vegetation.

For the land-cover classification, eleven classes were considered: asbestos roofs; asphalt; bare soil; concrete pavement; dark ceramic roofs; low vegetation; light ceramic roofs; metallic roofs; shadows; and water bodies. The input image was acquired in 2013 by the GeoEye-1 sensor (Figura 24). The image has four spectral bands (blue, green, red and infra-red), each with 0.5×0.5m pixel resolution. The image dimensions in pixels are 19,404×21,360 pixels.

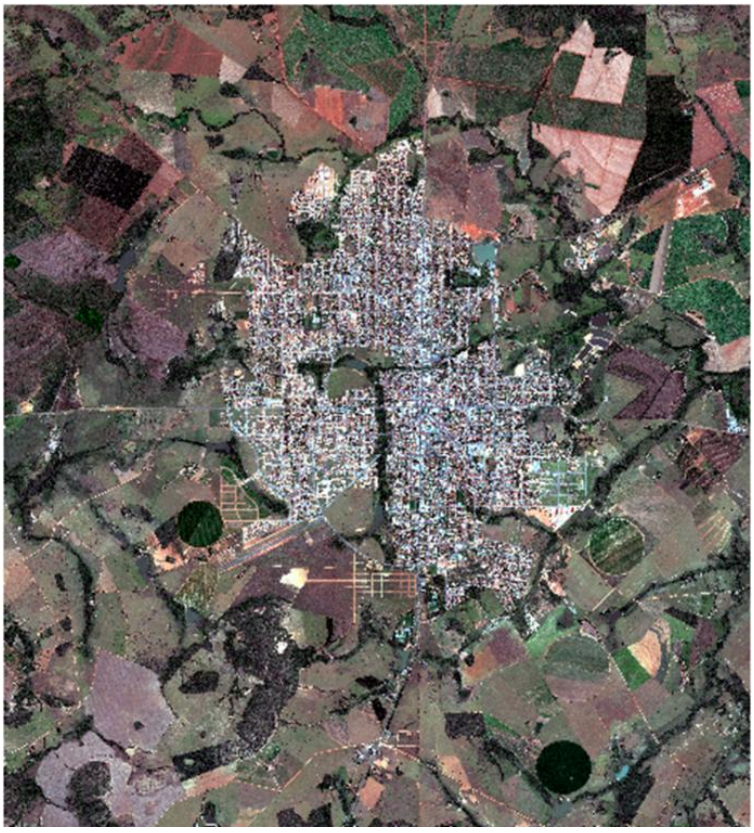


Figura 24 - RGB composition of the GeoEye-1 image used in the experiments.

C. Segmentation

Distributed segmentation was performed with the InterCloud system, using the hierarchical post-processing with re-segmentation technique (see Section III.A). The tile size was set to 512×512 pixels, resulting in a total of 1,677 tiles.

The region-growing segmentation parameter values were selected empirically as: 60 (scale); 0.5 (color weight); and 0.6 (compactness weight). The segmentation procedure running with those parameters generated a total of 239,156 segments.

D. Feature Computation

For each segment produced with the distributed segmentation procedure, the following 24 feature values were computed. Spectral features: brightness; mean intensity value in each of the four bands; maximum and minimum intensity value in each band; ratio intensity values of all bands; mean value in the infra-red band divided by the mean value in the blue band; and mean value in the infra-red band

divided by the mean value in the red band. Morphological features: area; mean angle; compactness; squareness; and roundness.

E. Sample Selection

After the segmentation step, a total of 510 sample/reference segments were carefully selected. A total of 50 samples were selected for all classes, but the concrete pavement and water bodies classes, for each of which 30 segments were selected. The reason for selection of less samples for these two classes was the scarcity of representative areas.

F. Classification Modeling

The method from the scikit-learn library used for generating the decision tree for classification was the DecisionTreeClassifier method. The method was executed having as inputs 75% of the samples for training, and 25% for testing.

RESULTS AND DISCUSSION

For thematic accuracy assessment, 765 random test points were defined randomly, with the aid of the QuantumGIS software. The number of test points was defined according to Congalton and Green [30]. The land-cover class associated with each test point was determined by visual inspection of the input image. Such assessment supported the construction of a confusion matrix (Figura 25), from which per class commission and omission error values were computed.

	Asbestos roof	Asphalt	Bare Soil	Concrete pavement	Dark ceramic roof	Low vegetation	Light ceramic roof	Metallic roof	Shadow	Water	Arboreal vegetation	Commission error (%)
Asbestos roof	**5**	4		4					1			**0.64**
Asphalt		**61**		2					7			**0.13**
Bare Soil	3	8	**143**	2	23	7	8					**0.26**
Concrete pavement	2	1		**16**				4				**0.30**
Dark ceramic roof	1		7	1	**28**	2						**0.28**
Low vegetation						**241**			1			**0.00**
Light ceramic roof			5			2	**22**					**0.24**
Metallic roof		3		2				**16**		2		**0.30**
Shadow		2				1			**11**			**0.21**
Water									2	**5**		**0.29**
Arboreal vegetation	1										**109**	**0.01**
Omission error (%)	**0.58**	**0.23**	**0.08**	**0.41**	**0.45**	**0.05**	**0.27**	**0.20**	**0.50**	**0.29**	**0.00**	

Figura 25 - Classification confusion matrix.

The global accuracy obtained in the classification experiments was 0.86, which is considered a satisfactory result based on the number of classes and the classification algorithm used. The thematic map produced with the classification results is shown in Figura 26. Figura 27 shows a subset of the thematic map, together with the corresponding image subset.

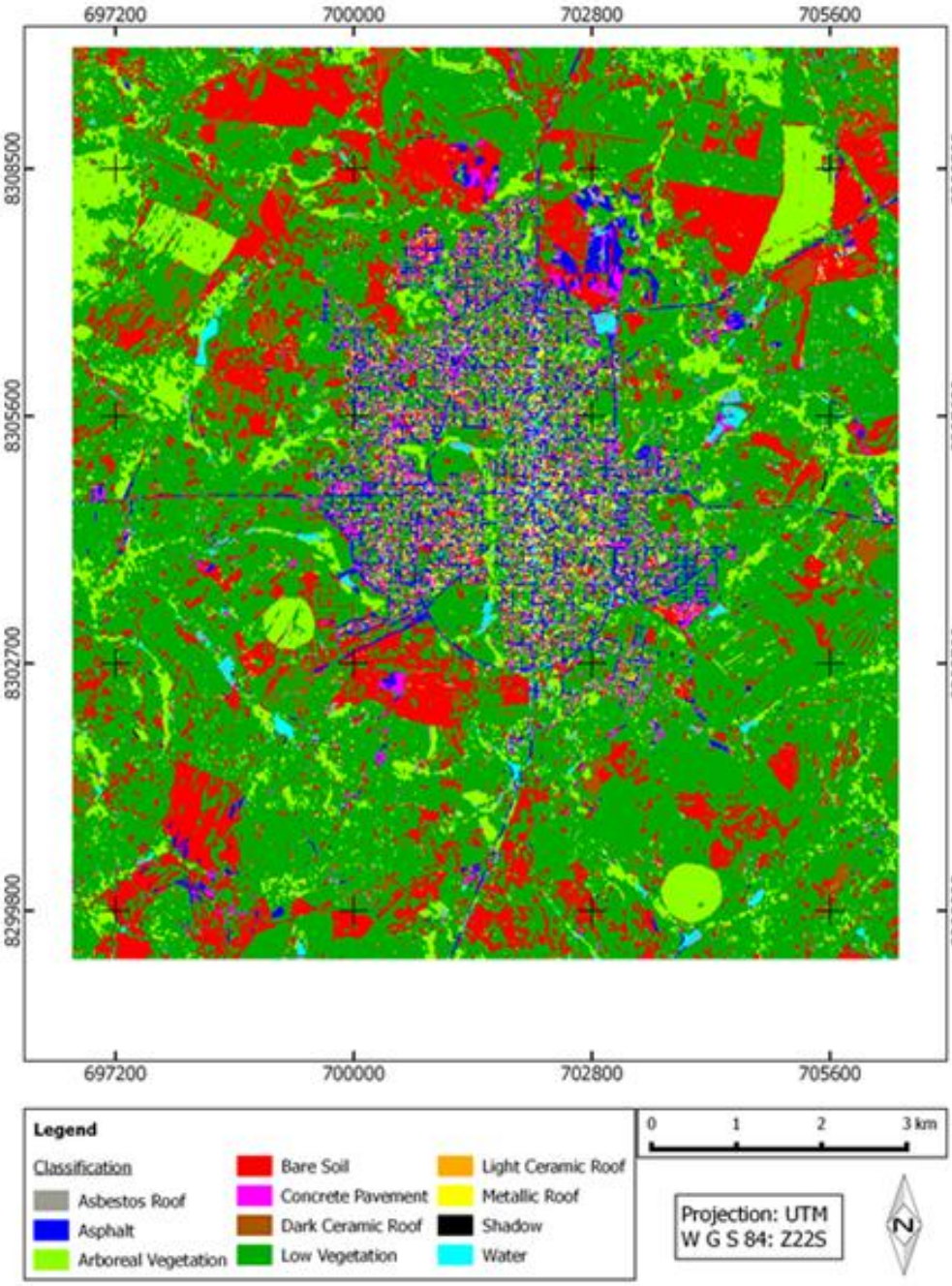


Figura 26 - Thematic map with the object-based classification outcome.

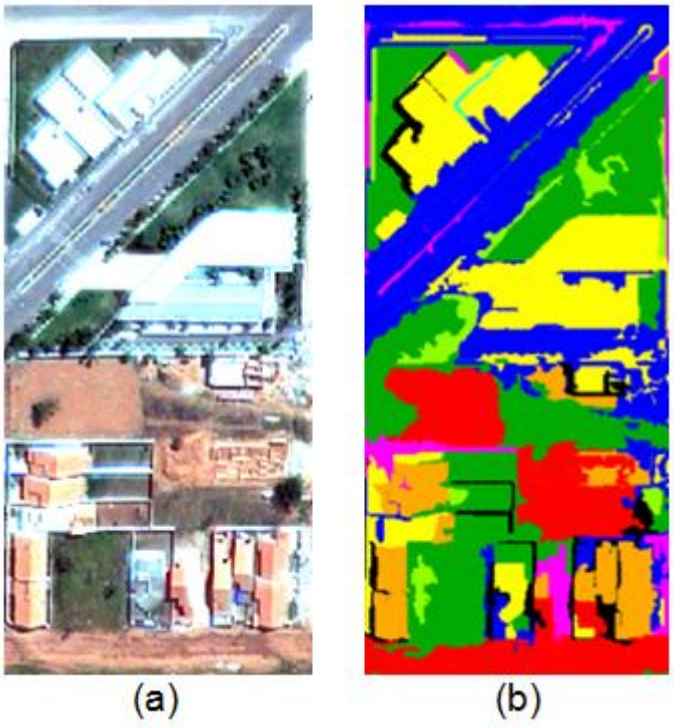


Figura 27 - Detailed close-up of the input image (a) and corresponding classification result (b).

In order to evaluate the scalability of the proposed approach, the batch distributed classification procedure, which comprises steps (a), (b) and (e) of the approach (see Figura 1), was executed six times, each time with a virtual cluster with different number of processing nodes: 2, 4, 8, 16 and 32. Figura 28 shows the processing times associated with each cluster configuration.

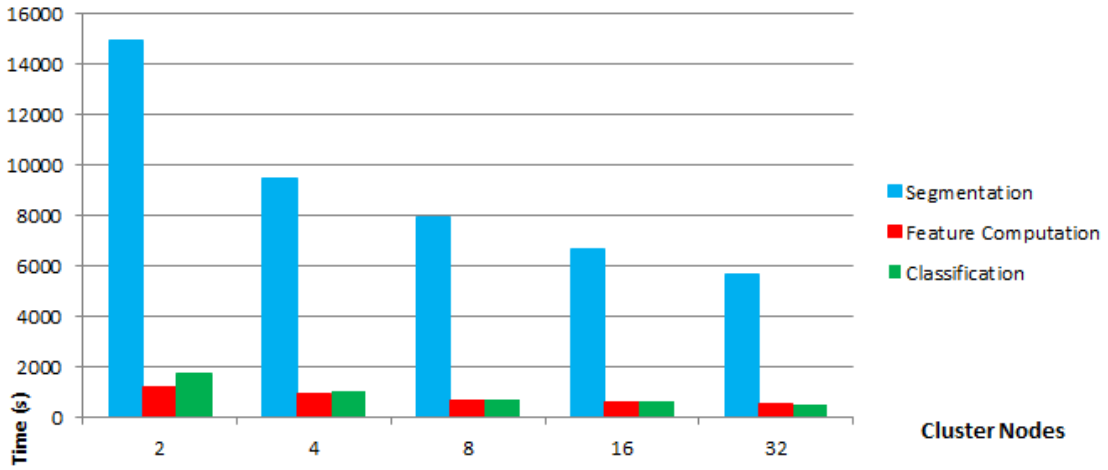


Figura 28 - Processing times associated with the segmentation, feature computation and classification steps of the proposed approach.

Figura 28 shows that the more expensive step in terms of time consumption is segmentation, which is also the step that benefits the most by scaling up cluster resources. The Figura also shows that the corresponding speedups do not increase linearly as the number of processing units increases. Such nonlinear effects are common in parallel systems: as more processing units are used, the workload of each unit decreases and the parallelization overhead becomes more pronounced.

It is important to observe that regardless of the virtual cluster used, the exact same segmentation and classification results were achieved, which shows the robustness of the approach with respect to scaling cluster resources.

CONCLUSION

In this work, we introduced a cloud computing approach for object-based image analysis and classification of arbitrarily large RS data sets. The approach enables machine learning methods to be exploited in the modeling of classification models, through the use of a web-based notebook system. The proposed approach was implemented with the methods available in the InterCloud system, which were integrated with the Apache Zeppelin notebook system, providing support for collaborative data analytics and visualization. Moreover, the Apache Zeppelin system provided the means for exploiting the scikit-learn Python machine learning when creating a particular classification model.

We evaluated the implementation of the proposed approach with a land-cover object-based image interpretation application, carried out over a large GeoEye-1 scene, using virtual clusters from a commercial cloud computing infrastructure service provided. The results obtained showed the effectiveness of the approach in efficiently handling a fairly large data volume in a scalable way, in terms of the number of allocated computing resources.

Although the application used for experimental evaluation consists of a straightforward supervised classification, we believe that through the use of the notebook system in a collaborative way the approach can be used to devise highly complex classification problems, combining machine learning and structural classification rules based on prior knowledge.

As future research, we envisage exploiting different machine learning methods, through the integration of alternative software libraries, combined with

fuzzy or crisp production rules to produce complex classification models. Finally, we would like to stress that the proposed approach was entirely implemented with open-source tools, thus providing a base for scientific collaboration and further research by other groups.

BIBLIOGRAPHIC REFERENCES

1. NASA EARTHDATA. EOSDIS Annual Metrics Reports. [Online]. https://earthdata.nasa.gov/about/system-performance/eosdis-annual-metrics-reports (accessed December 2017).

2. Lee, J. G.; Kang, M. Geospatial big data: challenges and opportunities. Big Data Research, vol. 2, pp. 74-81, June 2015.

3. Ma, Y.; Wu, H.; Wang, L.; Huang, B.; Ranjan, R.; Zomaya, A.; and Jie, W. Remote sensing big data computing: Challenges and opportunities. Future Generation Computer Systems, 51, 47-60., October 2015.

4. Jadhav, D. K. "Big Data: The New Challenges in Data Mining," International Journal of Innovative Research in Computer Science and Technology, vol. 1, no. 2, pp. 39-42, September 2013.

5. Fernández, A.; del Río, S.; López, V.; Bawakid, A.; del Jesus, M. J.; Benítez, J. M., and Herrera, F.; “Big Data with Cloud Computing: an insight on the computing environment, MapReduce, and programming frameworks,” Wiley Interdisciplinary Reviews: Data Mining and Knowledge Discovery, vol. 4, nº 5, pp. 380-409, September/October 2014.

6. Ferreira, R.S.; Bentes, C.; Costa, G.A.O.P.; Oliveira, D.A.B.; Happ, P.N., Feitosa, R.Q.; and Gamba, P. “A Set of Methods to Support Object-Based Distributed Analysis of Large Volumes of Earth Observation Data,” IEEE Journal of Selected Topics in Applied Earth Observations and Remote Sensing (JSTARS), vol. 10, no. 2, pp. 681–690, 2017.

7. Blaschke T., “Object based image analysis for remote sensing,” ISPRS J. Photogramm. Remote Sens., vol. 65, no. 1, pp. 2–16, Jan. 2010.

8. Blaschke, T.; Hay, G. J.; Kelly, M.; Lang, S.; Hofmann, P.; Addink, E.; Feitosa, R.; Meer, F.; Werff, H.; Coillie, F. and Tiede, D. (2014). Geographic object-based image analysis–towards a new paradigm. ISPRS Journal of Photogrammetry and Remote Sensing, 87, 180-191.

9. Costa, G.A.O.P.; Ferreira R.S.; Bentes, C.; Feitosa, R.Q.; and Oliveira, D.A.B., “Exploiting Different Types of Parallelism in Distributed Analysis of Remote Sensing Data,” IEEE Geoscience and Remote Sensing Letters, vol. 14, no. 8, pp. 1298–1302, 2017.

10. Happ, P. N.; Costa , G. A. O. P.; Bentes, C.; Feitosa, RQ.; Ferreira, R.S.; and Farias, R. “A cloud computing strategy for region-growing segmentation,” IEEE J. Sel. Topics Appl. Earth Observ. Remote Sens., vol. 9, no. 12, pp. 5294–5303, Dec. 2016.

11. Quirita, V. A. A.; Costa, G. A. O. P.; Happ, P. N.; Feitosa, R. Q.; Ferreira, R. D. S.; Oliveira, D. A. B.; and Plaza, A., “A new cloud computing architecture for the classification of remote sensing data,” IEEE Journal of Selected Topics in Applied Earth Observations and Remote Sensing, vol. 10, no. 2, pp. 409–416, 2017.

12. Dean, J.; and Ghemawat, S. “MapReduce: Simplified data process on large clusters,” Commun. ACM, vol. 51, no. 1, pp. 107–113, 2008.

13. Apache Zeppelin. https://zeppelin.apache.org/ (accessed December 2017).
14. Scikit-learn. Documentation. http://scikit-learn.org/stable/documentation.html (accessed December 2017).

15. Golpayegani, N.; and Halem, M. “Cloud computing for satellite data processing on high end compute clusters,” in Proc. IEEE Int. Conf. Cloud Comput., 2009, pp. 88–92.

16. Chen, Z.; Chen, N.; Yang, C.; and Di, L. “Cloud computing enabled web processing service for earth observation data processing,” IEEE J. Sel. Topics Appl. Earth Observ. Remote Sens., vol. 5, no. 6, pp. 1637–1649, Dec. 2012.

17. Cappelaere, P.; Sanchez, S.; Bernab S.; Scuri, A.; Mandl, D.; and Plaza, A. “Cloud implementation of a full hyperspectral unmixing chain within the NASA web coverage processing service for EO-1,” IEEE J. Sel. Topics Appl. Earth Observ. Remote Sens., vol. 6, no. 2, pp. 408–418, Apr. 2013.

18. Tan, X.; Di, L.; Deng, M.; Chen, A.; Huang, F.; Peng, C.; Gao, M.; Yao, Y.;and Sha, Z. “Cloud-and agent-based geospatial service chain: A case study of submerged crops analysis during flooding of the yangtze river basin,” IEEE J. Sel. Topics Appl. Earth Observ. Remote Sens., vol. 8, no. 3, pp. 1359–1370, Mar. 2015.

19. Lv, Z.; Hu, Y.; Zhong, H.; Wu, J; Li, B.; and Zhao, H. “Parallel k-means clustering of remote sensing images based on MapReduce,” in Proc. Int. Conf. Web Inf. Syst. Mining, 2010, pp. 162–170.

20. Liu, Y.; Chen, L.; Xiong, W.; Liu, L.;and Yang, D. “A MapReduce approach for processing large-scale remote sensing images,” in Proc. 20th Int. Conf. Geoinformat., 2012, pp. 1–7.

21. Almeer, M. H.; “Cloud Hadoop map reduce for remote sensing image analysis,” J. Emerg. Trends Comput. Inf. Sci., vol. 3, no. 4, pp. 637–644, 2012.
22. Cary, A.; Sun, Z.; Hristidis, V.; and Rishe, N.; “Experiences on processing spatial data with MapReduce,” in Proc. 21st Int. Conf. Sci. Statist. Database Manage., 2009, pp. 302–319.

23. Lu, J.; and Guting, R. H. “Parallel secondo: Boosting database engines with Hadoop,” in Proc. 18th Int. Conf. Parallel Distrib. Syst., 2012, pp. 738– 741.

24. Zhong, Y.; Han, J.; Zhang, T; Li, Z; Fang, J; and Chen, G. “Towards parallel spatial query processing for big spatial data,” in Proc. 26th Int. Parallel Distrib. Process. Symp. Workshops Ph.D. Forum, 2012, pp. 2085–2094.

25. Aji, A.; Wang, F.; Vo, H.; Lee, R.; Zhang, X.; and Saltz, J. “Hadoop GIS: A high performance spatial data warehousing system over MapReduce,” Proc. VLDB Endowment, vol. 6, no. 11, pp. 1009–1020, 2013.

26. Eldawy, A.; and Mokbel, M. F. “Spatialhadoop: A mapreduce framework for spatial data,” in Proc. 31st Int. Conf. Data Eng., 2015, pp. 1352–1363.

27. Gorelick, N.; Hancher, M.; Dixon, M.; Ilyushchenko, S.; Thau, D.; Moore, R. “Google Earth Engine. Planetary-scale geospatial analysis for everyone”. Remote Sensing of Environment 202 (Supplement C), 2017, pp. 18–27.

28. Aggarwal, C. Data Classification: Algorithms and Applications. New York, USA: Chapman and Hall/CRC, 2015.

29. A. Gates, Programming Pig. Sebastopol, CA, USA: O’Reilly Media, 2011.

30. Congalton, R. G.; Green, K. Assessing the accuracy of remotely sensed data: principles and practices. Boca Raton-USA: Lewis Publisher, 1999.

5 - RESULTADOS E DISCUSSÃO

INTEGRAÇÃO DE FERRAMENTAS DE CÓDIGO LIVRE PARA CLASSIFICAÇÃO OBIA – *DESKTOP*

Como parte do resultado desta pesquisa, mostrou-se que outros mineradores de dados, além do WEKA, também podem ser utilizados na integração de um processo de classificação baseado em OBIA utilizando o sistema classificador InterIMAGE. Os mineradores testados para integração com InterIMAGE nesta pesquisa foram: Orange Canvas, SIPINA, KNIME e RapidMiner. Cada um com características próprias de operação e variedade de algoritmos. Os resultados das classificações foram avaliados por meio dos índices Global, Kappa e Tau e são apresentados nos artigos 1 e 2 (seção 2 e 3).

INTEGRAÇÃO DE FERRAMENTAS DE CÓDIGO LIVRE PARA CLASSIFICAÇÃO OBIA – EM NUVEM

Neste trabalho, foi introduzida abordagem de integração de ferramentas *open souce* para processamento distribuído, em nuvem, para análise de imagem baseada em objetos e classificação de imagem de sensoriamento remoto de grande tamanho (19.404 x 21.360 pixels). A proposta foi implementada com os métodos disponíveis no sistema InterCloud, integrando-o com o sistema de *notebook web* Apache Zeppelin, que possibilitou o suporte para análise de dados colaborativos e visualização dos procedimentos executados. Vale destacar que a plataforma inicial do InterCloud não possui essa possibilidade, ficando o operador à mercê do resultado fornecido, gerando grande retrabalho.

Além disso, o sistema Apache Zeppelin forneceu meios para explorar as bibliotecas de aprendizagem de máquina Python Learning e Mllib Spark na criação de modelo de classificação específico ou disponibilizar consultas SQL.

5.2.1 Segmentação distribuída

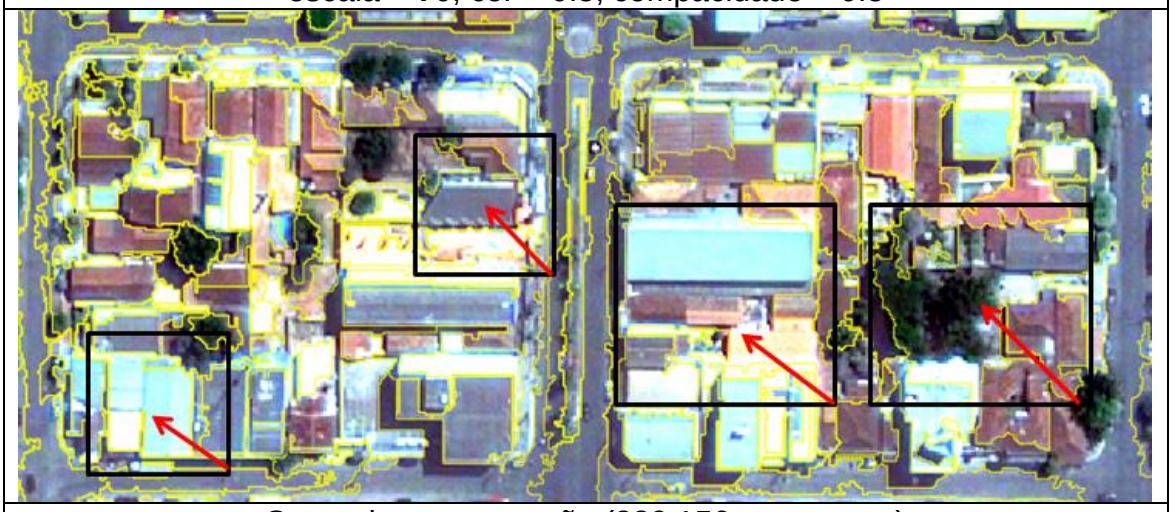
O algoritmo de Baatz e Shape (2000) foi utilizado para a segmentação multi-resolução.

No InterCloud foi estabelecido o tamanho do *tile* de 512 x 512, que resultou em 1.677 *tiles* e 239.156 segmentos. Foi necessário ajustar os parâmetros, porque em um primeiro processo de segmentação, usando escala maior (70), vários objetos não estavam corretamente rotulados, como mostrado na Figura 30.

Figura 29 - Segmentos gerados pelo InterCloud.

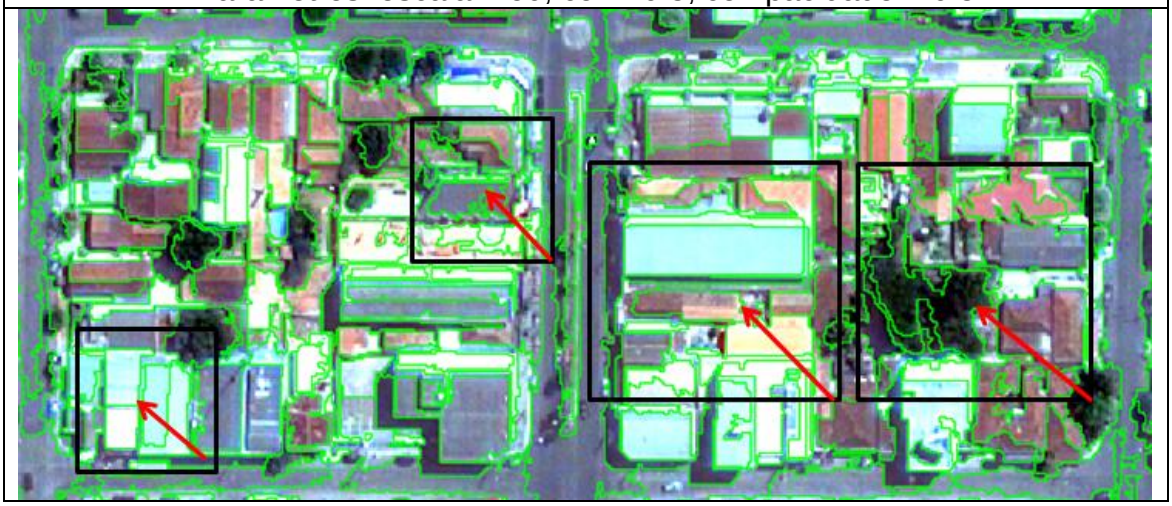
Primeira Segmentação (178.720 segmentos)

escala = **70**, cor = 0.5, compacidade = 0.5



Segunda segmentação (239.156 segmentos)

Parâmetros: escala = **60**, cor = 0.5, compacidade = 0.5



Fonte: Elaborado pelo autor.

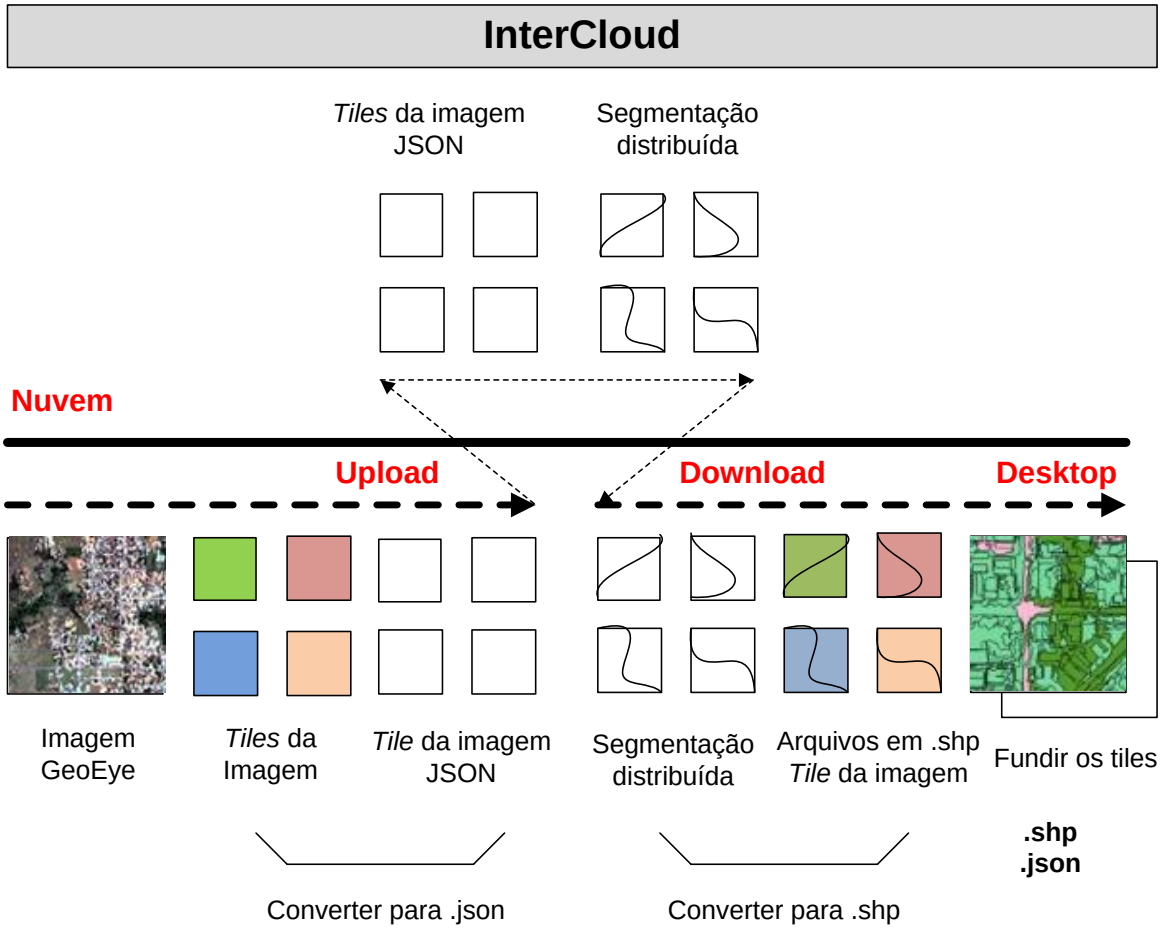
É possível observar os melhores objetos rotulados (setas vermelhas) na segunda segmentação, devido ao melhor ajuste do parâmetro de escala (60). Atualmente, no InterCloud, a questão de ajustes de parâmetros ainda é difícil e demorada porque existem vários processos de configuração manual na linha de código e procedimentos como *upload/download* de *scripts* (Pig) e conversão de .json para .shp, que é demorada e retarda o processo. Esses procedimentos são

totalmente diferentes daqueles utilizados no InterIMAGE, no qual a segmentação pode ser facilmente simulada com diferentes parâmetros até o ajuste ideal dos objetos.

O resultado da segmentação distribuída do InterCloud com 1.677 *tiles* foi armazenado em diretório na nuvem (HDFS Hadoop) e procedimentos para conversão e fusão dos *tiles* foram executados conforme apresentado na Figura 31. As conversões de .json e .shp foram executadas por módulos do InterCloud CreateShape e CreateJson e para a fusão foi utilizado o *software* QuantumGIS, por meio da ferramenta MergeShapes. O resultado final é uma camada com toda a imagem segmentada e disponibilizada em .shp e .json.

A execução dos procedimentos de conversão e *upload/download* de arquivos (.json e .shp) foi considerada fácil, porém demorada. Para agilizar esses procedimentos, foi contratado junto à provedora (AWS - Amazon Web Services Cloud) serviço (S3 Transfer Acceleration) que possibilita as transferências de arquivos rápidas para ambiente de computação distribuída.

Figura 30 – Procedimentos de conversão .shp e .json



Fonte: Elaborado pelo autor.

A camada resultante foi utilizada nas tarefas posteriores de extração de características e classificação distribuída baseada em objetos.

5.2.2 Extração das características

Para a extração das características, 24 atributos foram disponibilizados no InterCloud:minPixVal1,minPixVal2,minPixVal3,minPixVal4,bandDiv43,bandDiv41,ratio1,ratio2,ratio3,ratio4,angle,mean1,mean2,mean3,mean4,maxPixVal1,maxPixVal2,maxPixVal3,maxPixVal4,rectangle,roundness,compactness,brightness,area.

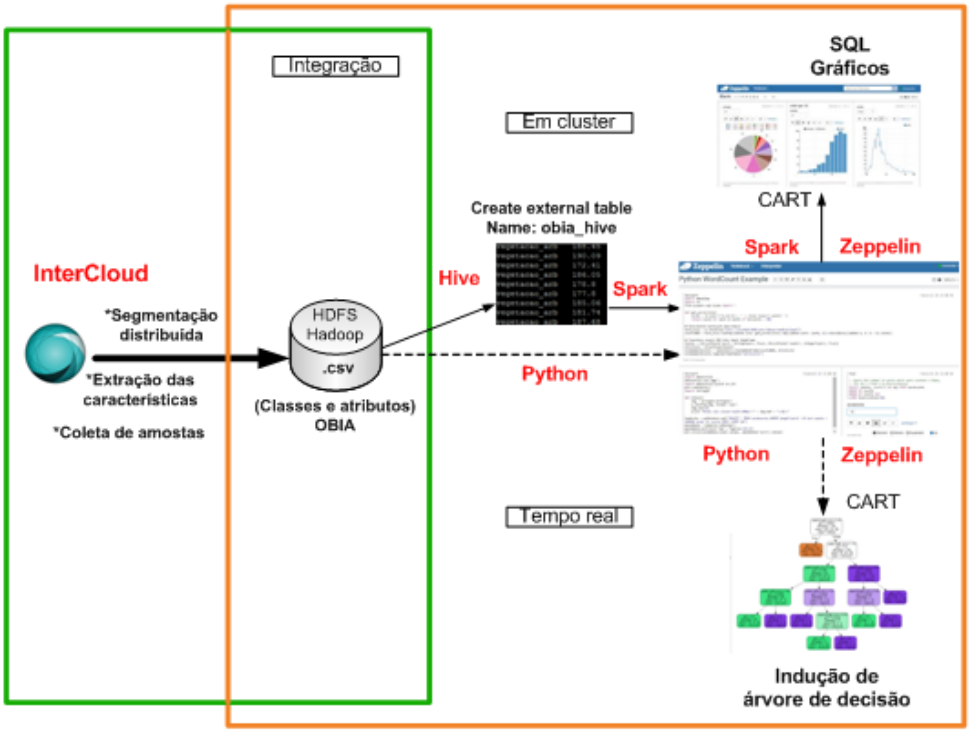
Atributos como entropia, soma de pixels e variância ainda não estão disponíveis e dependem de novas implementações. O resultado da segmentação, que pode ser considerado a principal tarefa do processo de pré-classificação, foi utilizado posteriormente na tarefa de extração das características.

5.2.3 Classificação supervisionada em nuvem

O método de classificação supervisionada no InterCloud está atualmente implementada usando apenas funções do WEKA (*Waikato Environment for Knowledge Analysis*), o que torna o sistema totalmente dependente dos algoritmos existentes daquele sistema minerador de dados. O método atual é baseado exclusivamente na execução de algoritmos de aprendizado de máquina, não há visualização gráfica (interface gráfica) dos dados classificados e nem como criar modelo de interpretação.

Como resultado, esta pesquisa mostrou que outros *frameworks* (Apache Spark, Apache Hive e Apache Zeppelin) e biblioteca de aprendizado de máquinas (Python Learning e Spark MLlib) foram suportados para processo de classificação distribuída baseado em objetos. A Figura 32 apresenta o novo modelo da implementação e integração das novas ferramentas com o InterCloud.

Figura 31 - - Integração do InterCloud com novas ferramentas e bibliotecas de aprendizagem de máquina



Fonte: Elaborado pelo autor.

O Hive permitiu criar nova tabela virtual diretamente no *cluster*, com classes e atributos oriundos do conjunto de dados em formato .csv, armazenados em HDFS. A declaração e instrução na linguagem hiveQL são apresentadas na Figura 33.

Figura 32 - Declaração e instrução (Apache Hive)

```
CREATE EXTERNAL TABLE obia_hive
(compactness FLOAT,
brightness FLOAT,
area FLOAT,
minPixVal3 FLOAT,
minPixVal2 FLOAT,
minPixVal1 FLOAT,
roundness FLOAT,
bandDiv43 FLOAT,
bandDiv41 FLOAT,
minPixVal4 FLOAT,
ratio2 FLOAT,
ratio1 FLOAT,
rectangle FLOAT,
mean3 FLOAT,
mean4 FLOAT,
angle FLOAT,
mean1 FLOAT,
ratio4 FLOAT,
ratio3 FLOAT,
mean2 FLOAT,
maxPixVal1 FLOAT,
maxPixVal4 FLOAT,
maxPixVal2 FLOAT,
maxPixVal3 FLOAT,
class STRING)
ROW FORMAT DELIMITED
FIELDS TERMINATED BY ','
LINES TERMINATED BY '\n'
LOCATION 's3n://icpunb/interimage/DataMining/';
 MSCK REPAIR TABLE obia_hive;
```

Fonte: Elaborado pelo autor.

Os valores, em pixels, foram disponibilizados para visualização em forma de tabela, e uma primeira checagem nos dados para validação pode ser feita pelo analista. A Figura 34 apresenta exemplo da seleção da classe ConcretePavement e atributos da tabela virtual. É possível analisar na tabela possíveis "ruídos" e artefatos no conjunto de dados e também checar valores inconsistentes.

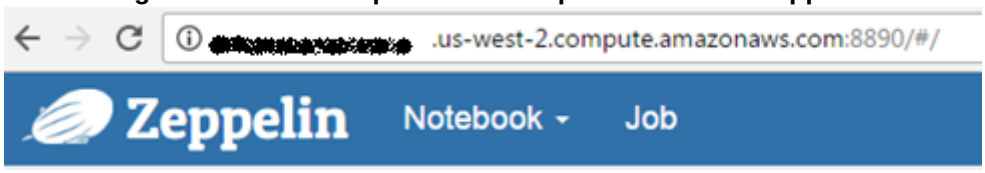
Figura 33 - Tabela virtual (Apache Hive)

```
Time taken: 0.178 seconds
hive> Select  class,mean3,mean4,angle
OK
ConcretePavement        392.74  688.12  1.57
ConcretePavement        337.44  692.05  1.43
ConcretePavement        312.62  556.3   0.42
ConcretePavement        367.08  668.51  1.57
ConcretePavement        336.55  694.04  1.57
ConcretePavement        347.59  653.42  1.54
ConcretePavement        388.81  754.62  0.98
ConcretePavement        305.51  658.83  1.57
ConcretePavement        370.85  718.4   0.59
```

Fonte: Elaborado pelo autor.

O Zeppelin foi conectado ao *cluster* para disponibilizar interface gráfica de interpretação de comandos e visualização dos resultados da classificação. A conexão foi feita por meio de extensão FoxyProxy do navegador Firefox, que permitiu conexão personalizada direto com o *cluster* via http (*Hypertext Transfer Protocol*). A porta utilizada para acesso ao Zeppelin é 8890, conforme apresentado na Figura 35.

Figura 34 - Conexão personalizada para acesso ao Zeppelin.



Fonte: Elaborado pelo autor.

Foi utilizada a biblioteca de aprendizagem de máquina Mllib Spark para classificação de árvores de decisão. A implementação permitiu o treinamento em arquitetura de computação distribuída com várias instâncias. O algoritmo de aprendizagem de máquina utilizado no código está disponível na biblioteca MLlib, por meio de APIs (Apache Spark, 2017). Foi possível combinar vários algoritmos em um único fluxo de trabalho (*Pipeline*).

O código criado na linguagem Scala (Figura 36), suportada pelo Spark, permitiu acessar os registros do conjunto de dados na tabela virtual em *cluster*.

Figura 35 - Código na linguagem Scala (Spark)

```
Linha 1 - import org.apache.spark.ml.classification.DecisionTreeClassifier
Linha 2 - import org.apache.spark.ml.feature.{StringIndexer, IndexToString, VectorIndexer, VectorAssembler}
Linha 3 - import org.apache.spark.ml.evaluation.MulticlassClassificationEvaluator
Linha 4 - import org.apache.spark.sql.functions._
Linha 5 - import org.apache.spark.sql.Row
Linha 6 - import org.apache.spark.sql.types._
Linha 7 - val sqlContext = new org.apache.spark.sql.hive.HiveContext(sc)
Linha 8 - val dfraw = sqlContext.sql("select compactness,brightness,area,minPixVal3,minPixVal2,minPixVal1,roundness,bandDiv43,bandDiv41,
minPixVal4,ratio2,ratio1,rectangle,class,mean3,mean4,angle,mean1,ratio4,ratio3,mean2,maxPixVal1,
maxPixVal4,maxPixVal2,maxPixVal3 from obia_hive")
Linha 9 - val df = dfraw.select(
Linha 10 - $"compactness".as("compactness").cast(FloatType),
Linha 11 - $"brightness".as("brightness").cast(FloatType),
Linha 13 - $"area".as("area").cast(FloatType),
Linha 14 - $"minPixVal3".as("minPixVal3").cast(FloatType),
Linha 15 - $"minPixVal2".as("minPixVal2").cast(FloatType),
Linha 16 - $"minPixVal1".as("minPixVal1").cast(FloatType),
Linha 17 - $"roundness".as("roundness").cast(FloatType),
Linha 18 - $"bandDiv43".as("bandDiv43").cast(FloatType),
Linha 19 - $"bandDiv41".as("bandDiv41").cast(FloatType),
Linha 20 - $"minPixVal4".as("minPixVal4").cast(FloatType),
Linha 21 - $"ratio2".as("ratio2").cast(FloatType),
Linha 22 - $"ratio1".as("ratio1").cast(FloatType),
Linha 23 - $"rectangle".as("rectangle").cast(FloatType),
Linha 24 - $"class".as("class").cast(StringType),
Linha 25 - $"mean3".as("mean3").cast(FloatType),
Linha 26 - $"mean4".as("mean4").cast(FloatType),
Linha 27 - $"angle".as("angle").cast(FloatType),
Linha 28 - $"mean1".as("mean1").cast(FloatType),
Linha 29 - $"ratio4".as("ratio4").cast(FloatType),
Linha 30 - $"ratio3".as("ratio3").cast(FloatType),
Linha 31 - $"mean2".as("mean2").cast(FloatType),
Linha 32 - $"maxPixVal1".as("maxPixVal1").cast(FloatType),
Linha 33 - $"maxPixVal4".as("maxPixVal4").cast(FloatType),
Linha 34 - $"maxPixVal2".as("maxPixVal2").cast(FloatType),
Linha 35 - $"maxPixVal3".as("maxPixVal3").cast(FloatType)
Linha 36 - )
Linha 35 - df.printSchema()
Linha 35 - df.show(510)
```

Fonte: Elaborado pelo autor.

A linha 7 (Figura 36) representa o ponto de entrada (SQLcontext) para trabalhar com dados estruturados (linhas e colunas) no Apache Spark. O SQLcontext permitiu a criação de objetos DataFrame, bem como a execução de consultas SQL. As linhas 8 e 9 (Figura 36) são as declarações de todos os atributos e classes da tabela virtual, que são do tipo *Float* e *String*.

Algoritmos e tarefas de avaliação necessárias para compor o fluxo do trabalho (APACHE SPARK, 2017) foram importados do MLlib Spark, conforme apresentados nas linhas de 1 a 6 e descritos na Figura 37.

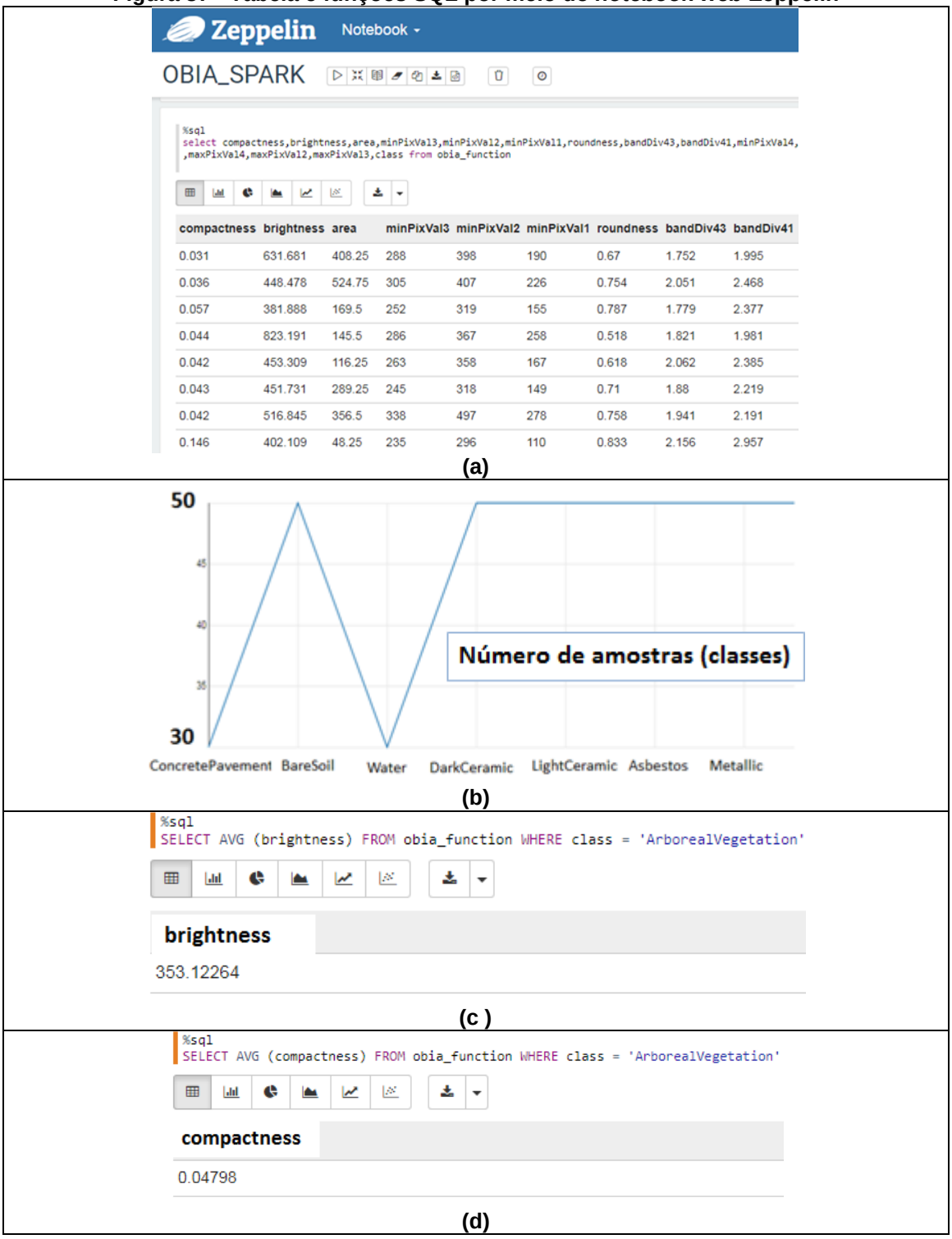
Figura 36 - Tarefas importadas da biblioteca MLlib Spark.

Linha 1 - org.apache.spark.ml.classification.DecisionTreeClassifier
Algoritmo de aprendizagem de árvore de decisão para classificação. Suporta classificação binárias e multiclass, bem como recursos contínuos e categóricos.
Linha 2 - org.apache.spark.ml.feature.{StringIndexer, IndexToString, VectorIndexer, VectorAssembler}
Algoritmos para extrair, transformar e selecionar recursos
Linha 3 - org.apache.spark.ml.evaluation.MulticlassClassificationEvaluator
Avaliador para classificação multiclass, que espera duas colunas de entrada: previsão e rótulo.
Linha 4 - org.apache.spark.sql.functions
Define funções do dataframe
Linha 5 - org.apache.spark.sql.Row
Representa uma linha de saída de um operador relacional.
Linha 6 org.apache.spark.sql.types._
Tipos de declarações. Exemplo: FloatType

Fonte: Elaborado pelo autor.

Com o código apresentado foi possível treinar um classificador de árvore de decisão e gerar valores de pixels previstos em cada registro no conjunto de dados. Com o resultado do novo método proposto é possível criar modelos estatísticos de interpretação e conhecimento para classificação (OBIA) com InterCloud. Além de gerar uma variedade de gráficos, pode-se, por exemplo, criar funções (SQL) ou até mesmo definir valor médio de pixel para cada classe/atributo, como mostrado na Figura 38, em que (a) é a tabela virtual com classes e atributos, (b) o gráfico quantitativo das amostras, e (c) e (d) são exemplos de função SQL simples para calcular e retornar o valor médio dos atributos de brilho e compacidade para a classe de vegetação arbórea.

Figura 37 - Tabela e funções SQL por meio do *notebook web* Zeppelin



Fonte: Elaborado pelo autor.

No *notebook web* Zeppelin, por meio de intérprete e biblioteca Python, foi possível executar código para classificar, induzir e visualizar a árvore de decisão por meio do algoritmo CART. O código foi desenvolvido na linguagem Python (Figura

39) e permitiu o acesso a registros do conjunto de dados armazenado no Hadoop HDFS. Todo o processo foi realizado diretamente no *clusters*.

Figura 38 - Código em Python para classificação supervisionada na nuvem

```
Linha 1 - import numpy as np
Linha 2 - import matplotlib.pyplot as plt
Linha 3 - import pandas as pd
Linha 4 ---
Linha 5 - dataset = pd.read_csv("s3://icpunb/interimage/DecisionTree/tree_.csv")
Linha 6 - X = dataset.iloc[:, :24].values
Linha 7 - y = dataset.iloc[:, -1].values
Linha 8 ---
Linha 9 - from sklearn.cross_validation import train_test_split
Linha 10 - X_train, X_test, y_train, y_test = train_test_split(X, y, test_size = 0.25, random_state = 0)
Linha 11 ---
Linha 12 - from sklearn.preprocessing import LabelEncoder, OneHotEncoder
Linha 13 - labelencoder_y = LabelEncoder()
linha 14 - y_train = labelencoder_y.fit_transform(y_train)
linha 15 - y_test = labelencoder_y.fit_transform(y_test)
linha 16 - y = labelencoder_y.fit_transform(y)
linha 17 ---
linha 18 - from sklearn.tree import DecisionTreeClassifier, export_graphviz
linha 19 - classifier = DecisionTreeClassifier(criterion = 'entropy', random_state = 0)
Linha 20 - classifier.fit(X_train, y_train)
Linha 21 ---
Linha 22 - import pydot
Linha 23 - dotfile = open("/tmp/rodrigo_tree", 'w')
Linha 24 - export_graphviz(classifier, out_file=dotfile,
Linha 25 - feature_names=dataset.iloc[:, :24].columns, class_names=labelencoder_y.classes_,filled=True) dotfile.close()
Linha 26 ---
Linha 27 - dot -Tpdf /tmp/rodrigo_tree -o treePdf.pdf
```

Fonte: Elaborado pelo autor.

A descrição de cada linha de código em Python é mostrada na Figura 40.

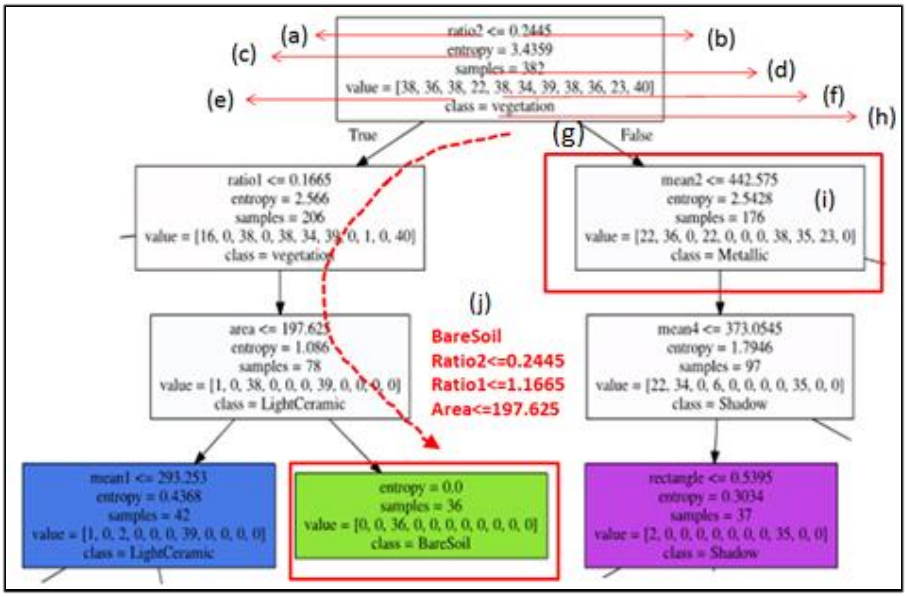
Figura 39 - Descrição do código em Python para classificação supervisionada na nuvem

Linha 1 - import numpy as np import pandas as pd
Importa um pacote da linguagem Python que permite trabalhar vetores e matrizes de N dimensões
Linha 2 - import matplotlib.pyplot as plt
Importa uma coleção de funções Python. Exemplo: criar figuras e traçar linhas
Linha 3 - import pandas as pd
Biblioteca manipulação e análise de dados em Python
Linha 5, 6 e 7 - dataset = pd.read_csv ("s3://icpunb/interimage/DecisionTree/tree_.csv") X = dataset.iloc[:, :24].values y = dataset.iloc[:, -1].values
Importando o conjunto de dados HDFS Hadoop (S3) com 24 classes para classificação
Linhas 9 e 10 - from sklearn.cross_validation import train_test_split X_train, X_test, y_train, y_test = train_test_split(X, y, test_size = 0.25, random_state = 0)
Divisão do conjunto de dados do treinamento (padrão 0.25)
Linhas 12,13,14, 15 e 16 - from sklearn.preprocessing import, labelencoder_y =LabelEncoder()LabelEncoder y_train = labelencoder_y.fit_transform(y_train) y_test = labelencoder_y.fit_transform(y_test) y=labelencoder_y.fit_transform(y)
LabelEncoder é uma utilidade de classes. Ajudar a normalizar rótulos de modo que eles contenham apenas valores entre 0 e n_classes-1
Linhas 18, 19 e 20 - from sklearn.tree import DecisionTreeClassifier, export_graphviz classifier = DecisionTreeClassifier(criterion = 'entropy', random_state = 0) classifier.fit(X_train, y_train)
Treinamento e representação gráfica da classificação de árvore de decisão. Entropia como critério de divisão.
Linhas 22,23,24 e 25 - import pydot dotfile = open("/tmp/rodrigo_tree", 'w') export_graphviz(classifier, out_file=dotfile, feature_names=dataset.iloc[:, :24].columns, class_names=labelencoder_y.classes_,filled=True) dotfile.close()
Pydot fornece uma interface para criar gráficos (dotfile) usando **graphviz.** É gerado um dotfile para criar o gráfico de árvore de decisão usando graphviz.
Linha 27 - dot -Tpdf /tmp/rodrigo_tree -o treePdf.pdf
Gera árvore de decisão em PDF. Comando executado no cluster master.

Fonte: Elaborado pelo autor.

O resultado, exportado em .pdf em diretório temporário no *cluster,* é uma grande árvore de decisão exploratória que apresentou 61 nós. Não foi necessário instalar ferramentas de *desktop* para a classificação da árvore de decisão e o código foi lido por meio de operação copiar/colar (*online)*. Parte da árvore é apresentada na Figura 41 como exemplo, mostrando cada item que compõe a estrutura da árvore. Nas Figuras (41a) e (41b), os exemplos de atributo (ratio2) e valor de limiar (<= 0.2445) da classe vegetação (41h) são mostrados no primeiro nó da árvore. Em (41c) é mostrado o valor da entropia, que corresponde a um valor de impureza do atributo. O número de amostra utilizado na primeira partição da classificação (com critério de 0,25) é apresentado em (41d). Cada valor apresentado no vetor em (41f) representa o número de amostras utilizadas em cada classe para o particionamento e a classificação de cada nó. Uma condição informativa (verdadeira ou falsa) é apresentada em (41g), na parte superior da árvore. Se a condição for verdadeira, segue o ramo esquerdo e, se falso, segue o ramo direito e a inversão na condição da regra é necessária quando o analista descreve a regra no roteiro do Script Pig do InterCloud. Um exemplo é dado em (41g), onde a regra representada pela condição negativa (falso) no ramo direito é descrita por mean2 <= 442.575 e deve ser convertida para a descrição Script Pig como mean2> 442.575.

Figura 40 - Parte do resultado da indução da árvore de decisão na nuvem (algoritmo CART)



Fonte: Elaborado pelo autor.

De acordo com Ferreira (2015), no InterCloud, os operadores e as regras de decisão são tratados da mesma forma e ambos estão escritos em Pig Latin. Um

operador do InterCloud é considerado um conjunto de operações de alto nível. Cada operação é representada por um conjunto de comandos de Pig Latin e UDFs que trabalham juntos para executar uma tarefa específica. O resultado da classificação de árvore de decisão (CART) deste trabalho resultou em um total de 32 regras, que foram simuladas e descritas em um *script* Pig do InterCloud para a classificação baseada em objetos.

O *script* Pig, com várias definições e funções, importou arquivo JSON armazenado em nuvem contendo os segmentos a serem classificados (OBIA). A mesma prioridade (1.0) foi dada para todas as regras, para cada classe. Uma simulação de uma regra (Figura 41j) descrita no *script* Pig é apresentada na Figura 42 para a classe solo exposto.

Figura 41 - Parte da descrição da regra do Script Pig do InterCloud

```
projection_1 = FOREACH projection_17 GENERATE geometry, data, ( CASE WHEN
(properties#'ratio2' <= 0.244) AND (properties#'ratio1' > 0.166) AND (properties#'area' >
197.625) THEN II_ToClassification('BareSoil', 1.0, properties) ELSE properties END ) AS
properties;
...
...
projection_32 = FOREACH projection_31 GENERATE geometry, data, ( CASE WHEN
(properties#'ratio2' > 0.244) AND (properties#'mean2' > 442.575) AND (properties#'ratio4' > 0.262)
AND (properties#'bandMeanDiv41' > 1.895) AND (properties#'maxPixVal3' <= 520.5) AND
(properties#'angle' > 0.813) AND (properties#'compactness' <= 0.010) THEN
II_ToClassification('Metallic', 1.0, properties) ELSE properties END ) AS properties;
```

Fonte: Elaborado pelo autor.

Os detalhes referentes aos resultados da classificação distribuída, como o mapa de cobertura do solo, tempo de processamento e acurácia, são apresentados no resultado do artigo 3 (seção 4).

5.2.4 Aplicação para classificação de imagens objetivando análise ambiental

Os resultados mostraram que o método proposto é uma alternativa viável para classificação de imagens, com a finalidade de aplicações de análise e monitoramento ambiental em grande escala, principalmente no que concerne ao processamento de grande volume de dados (aplicações BigData).

No que se refere ao desenvolvimento e aplicação de modelo de interpretação baseado em OBIA para análise ambiental (exemplo: monitoramento de grandes lagos, reservatórios de abastecimento, áreas mineradas, queimadas, desflorestamento, expansão urbana, etc.), o InterCloud ainda carece de operadores

(análises) considerados importantes para interpretação dos alvos. Um exemplo é a dificuldade de separação do solo exposto com cerâmica (mistura espectral), apresentado no resultado desta pesquisa (artigo 3, seção 3) e apresentado na Figura 43.

Figura 42 - Exemplo de mistura das classes (alvo)

Fonte: Elaborado pelo autor.

As dificuldades de separação ocorrem devido à presença das mesmas características: cor, textura e forma. Essas dificuldades de separação podem ser mais bem interpretadas pelo InterCloud com a implementação e aplicação de análise de contexto, vizinhança e distância.

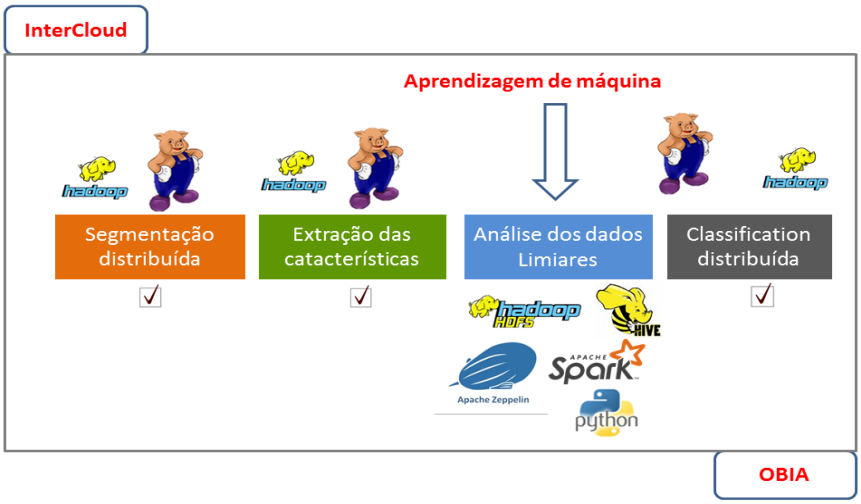
6 - CONCLUSÃO E RECOMENDAÇÕES

O sistema de código livre para interpretação automática de imagem InterIMAGE possui alguns operadores OBIA para classificação e apresenta-se como ótima opção para projetos de pesquisa em países que apresentam limitações expressivas nos investimentos para utilização de *softwares* proprietários (por exemplo: Brasil), como é o caso do eCognition, que possui custo elevado por licença, mesmo para fins educacionais. Contudo, a limitação do tamanho da imagem (3.000 x 3.000) e a ausência de operadores para análises que são essenciais em OBIA ainda não estão disponíveis no InterIMAGE, como por exemplo: análise de vizinhança, de contexto e distâncias. Conclui-se que a implementação de operadores para essas análises traria grande melhoria para o sistema e consequentemente para a classificação OBIA. Porém, percebe-se a falta de investimento nesse eficiente sistema, que carece de manutenção e atualização continuada. Sua última versão é de 2014.

Em relação à integração com outros mineradores de dados, além do WEKA, como por exemplo, o Orange Canvas e SIPINA, devem ser realizadas implementações, principalmente com a versão *Desktop* (InterIMAGE).

Nesta pesquisa foi apresentado novo método de integração de ferramentas de código livre para processos de classificação distribuída de objetos. O novo método é considerado módulo independente do modelo atual de aprendizagem de máquina do InterCloud, como apresentado na Figura 44.

Figura 43 - Novas ferramentas para o processo de classificação



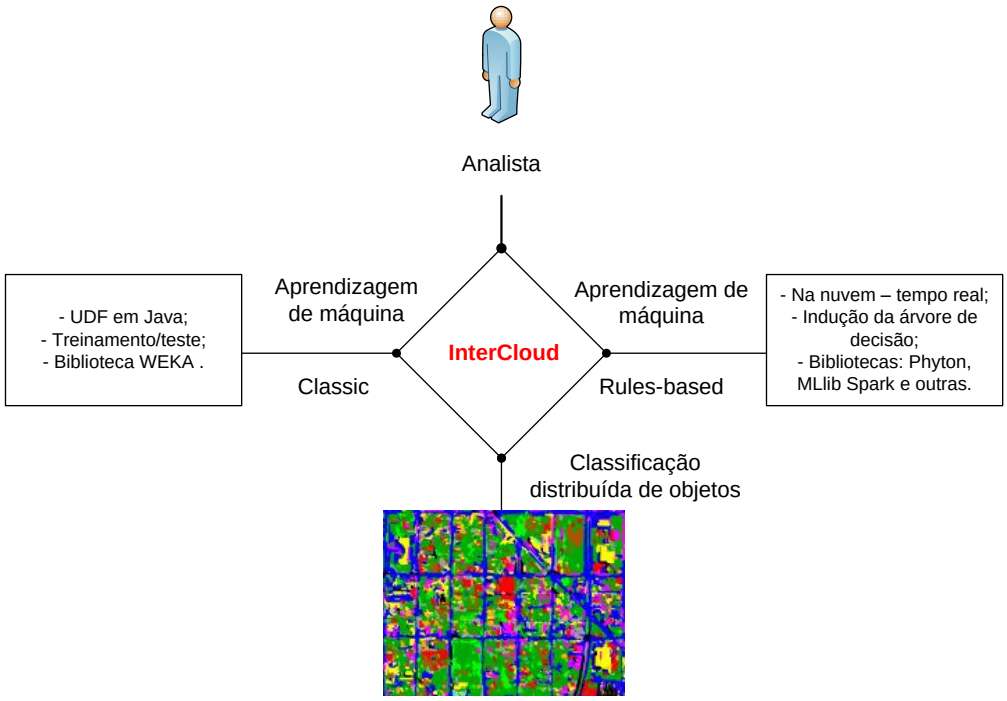
Fonte: Elaborado pelo autor.

Até agora, no InterCloud, o modelo de classificação supervisionada era baseado apenas nas funções do WEKA. Com o novo método proposto neste trabalho, outras estruturas e bibliotecas de aprendizado de máquina, como Python Learning e MLlib Spark, surgem como opção para os processos de classificação distribuída de objetos.

O código em Python e em Scala (MLlib Spark), executados diretamente *no cluster* (nuvem), tornou-se o processo mais ágil, sem a dependência de qualquer instalação de pacote de *software* no *desktop* ou mesmo de tarefas de *download* e *upload*. Outra ferramenta de código livre integrada com sucesso ao processo foi o *notebook Web* Apache Zeppelin para interpretação dos códigos e disponibilização de gráficos.

O novo modelo apresentado nesta pesquisa é considerado flexível e o analista que utilizará o InterCloud em seu projeto poderá optar por diferentes modelos de classificação (modelo clássico ou baseado em regras), conforme apresentado na Figura 45. Outras linguagens de programação e bibliotecas de aprendizado de máquina também podem ser utilizadas no processo de aprendizagem de máquina, por exemplo: Mahout, R e outros.

Figura 44 - Modelo flexível para classificação supervisionada na nuvem



Fonte: Elaborado pelo autor.

Na etapa de pré-processamento, a tarefa de segmentação de multi-resolução distribuída, utilizando o InterCloud, mostrou eficiência ao permitir a segmentação de imagem grande em ambiente de processamento distribuído e permitiu o uso de diferentes valores para os parâmetros escala, cor e forma. Para a tarefa de extração característica, é necessário implementar novos cálculos de atributos considerados importantes em classificação de imagem, como entropia e soma de pixels. O uso do processo de coleta de amostras para representar cada classe de objeto identificada na imagem foi satisfatório, já que definiu e separou cada classe de interesse por meio de identificação visual feita pelo analista, mas nenhuma ferramenta para essa tarefa foi implementada no InterCloud até o momento, causando dependência dos sistemas GIS (QuantumGIS, ArcGis e outros) para essa função.

O InterCloud provou ser excelente sistema para classificar imagens de grande escala, mas as vantagens típicas que caracterizam OBIA, como contexto, vizinhança topológica e distâncias ainda não foram implementadas em seu conjunto de operadores. É preciso desenvolver interface para inserção e manipulação de dados em todos os processos (segmentação, extração de características, aprendizagem de máquina e classificação de objetos), porque até agora toda a entrada de dados é feita por meio de códigos-fonte. A integração (usuário x máquina) ainda depende do bom conhecimento do analista em ferramentas de desenvolvimento, por exemplo, o Eclipse IDE para Java, que é a linguagem de programação do InterCloud.

A tarefa de descrever manualmente as regras com valores de limiares no *script* Pig do InterCloud é lenta e depende fortemente da atenção do analista para não gerar erros descritivos de entrada de dados e desqualificar o resultado de classificação distribuída baseada em objetos. Esses erros descritivos podem ser: dados duplicados, regras inversas, valores incorretos e outros. Recomenda-se a automatização dessa fase do processo.

Em relação à precisão temática, o resultado geral foi considerado satisfatório, atingindo acurácia Global de 85%, conforme o artigo 3 (seção 4) desta tese.

7 – REFERÊNCIAS BIBLIOGRÁFICAS GERAIS

ANTUNES, R. R., Bias, E. S., Brites, R. S., e Costa, G. A. O. P. Desenvolvimento de técnica para monitoramento do cadastro urbano baseado na classificação orientada a objetos. Estudo de caso: Município de Goianésia, Goiás. **Revista Brasileira de Cartografia**, n. 67/2, p. 357-372. Brasília, 2014.

ANTUNES, R. R., Bias, E. S., Costa, G. A. O. e Brites, R. S. Object-based analysis for urban land cover mapping using the InterIMAGE and the SIPINA free software packages. **Bulletin of Geodetic Sciences**, Vol. 24, issue 1, 1-17, Jan-Mar, 2018.

ANTUNES, R. R., Bias, E. S., Brites, R. S., e Costa, G. A. O. P. **Integration of open-source tools for object-based monitoring of urban targets.** GEOBIA 2016: Solutions and Synergies. University of Twente Faculty of Geo-Information and Earth Observation (ITC), 2016.

APACHE SPARK. **Spark documentation.** 2017. Disponível em: <https://spark.apache.org//> Acesso em: 29 ago. 2017.

BIAS, E. S.,Antunes, R,R. Brites, R. S., e Costa, G. A. O. P. Application of Imagery Analysis Based on Objects as a Tool for Monitoring the Urban Cadastre in Small Municipalities. **International Geographic Object-Based Image Analysis Conference**, Thessaloniki, 2014.

BLASCHKE, T., Hay, G. J., Kelly, M., Lang, S., Hofmann, P., Addink, E., and Tiede, D. (2014). Geographic object-based image analysis–towards a new paradigm. **ISPRS journal of photogrammetry and remote sensing**, 87, 180-191.

BLASCHKE, T. Object based image analysis for remote sensing. **ISPRS Journal of Photogrammetry and Remote Sensing**. v. 65, n. 1. Elsevier, Canada, 2010.

BLASCHKE, T.; TOMLJENOVIĆ, I. LIDARS capes and OBIA. **ASPRS 2012 Annual Conference Sacramento**. California, 2012.

CHEN, Q.; CHEN, Y. Object-based Change Detection of WorldView-2 data for Urban Dynamic Monitoring. **South-Eastern European Journal of Earth Observation and Geomatics**. Aristotle University of Thessaloniki, Greece. v. 3, n. 2S, 2014.

COSTA, G. A. O. P., Feitosa, R. Q., Fonseca, L. M. G., Oliveira, D. A. B., Ferreira, R. S., and Castejon, E. F. (2010). **Knowledge-based interpretation of remote sensing data with the InterIMAGE System**: Major characteristics and recent developments. GEOBIA 2010. Gent, Belgium, 2010.

DE GRANDE, T. O.; DE ALMEIDA, T.; CICERELLI, R. E. Classificação orientada a objeto em associação às ferramentas reflectância acumulada e mineração de dados. **Pesquisa Agropecuária Brasileira,** v. 51, n. 12, 2017, p. 1983-1991.

DOS ANJOS,L., C., Almeida, C. M., Soares G., L., e Souza Filho, C. R.. Análise do nível de legenda de classificação de áreas urbanas empregando imagens multiespectrais e hiperespectrais com os métodos árvore de decisão c4.5 e floresta randômica. **Boletim de Ciências Geodésicas**, v. 23, n. 2, 2017, Curitiba, Brasil.

FARIA, M., M. de Souza, L., F., T., Persil, V., H., Filho, E., I. and Francelino M.R. Use of classification algorithms J48 for mapping land use and cover. **South-Eastern European Journal of Earth Observation and Geomatics**. Grécia: Aristotle University of Thessaloniki, 2014.

FERREIRA, R., S., Oliveira, D. A. B., Happ, P. N., da Costa, G. A. O. P., Feitosa, R. Q., and Bentes, C. InterImage Cloud Platform: em direção à arquitetura de uma plataforma distribuída e de código aberto para a interpretação automática de imagens baseada em conhecimento. **XVII Simpósio Brasileiro de Sensoriamento Remoto**. João Pessoa, 2015.

FURTADO, L. F. A., Silva, T. S. F., Fernandes, P. J. F., and Novo, E. M. L. D. M.. **Land cover classification of Lago Grande de Curuai floodplain (Amazon, Brazil) using multi-sensor and image fusion techniques.** Sielo - Scientific Electronic Library Online. São Paulo, 2015.

INTERIMAGE. **Manual do usuário**, 2010. Disponível em: <http://www.lvc.ele.puc-rio.br/projects/interimage/pt-br/documentacao/> Acesso em: 23 ago. 2017.

JADHAV, D. K. Big Data: the new challenges in data mining. **International Journal of Innovative Research in Computer Science and Technology,** v. 1, n. 2, p. 39-42, September 2013.

LANG, S. Object-based image analysis for remote sensing applications: modeling reality – dealing with complexity. In: BLASCHKE, T.; LANG, S.; HAY, G. J. **Object-based image analysis**. Berlim: Springer, 2008.

LACERDA, C. S. D. A., Almeida, C. M. D., Galvão, L. S., and Souza Filho, C. R. Analysis of the level of detail in classifications of urban areas with optical VHR and hyperspectral images using a nonparametric method. **South-Eastern European Journal of Earth Observation and Geomatics**. Grécia: Aristotle University of Thessaloniki, 2014.

LEE, J. G.; KANG, M. Geospatial big data: challenges and opportunities. **Big Data Research**, v. 2, p. 74-81, June 2015.

LIU, P. A survey of remote-sensing big data. **Frontiers in Environmental Science**. v. 3, 2015, p. 45.

MA, Y., Wu, H., Wang, L., Huang, B., Ranjan, R., Zomaya, A., and Jie, W. Remote sensing big data computing: Challenges and opportunities. **Future Generation Computer Systems**, v. 51, p. 47-60, October 2015.

NASA EARTHDATA. **EOSDIS Annual Metrics Reports**. 2017. Disponível em: <https://earthdata.nasa.gov/about/system-performance/eosdis-annual-metrics-reports> Acesso em: 20 dez. 2017.

NASCIMENTO, A. F. Rubim, I. B., Pereira, E. G. S., de Barros, R. S., and Richter, M.. Classificação da cobertura da terra, utilizando os programas livres: InterIMAGE, WEKA e QuantumGIS. **Anais XVI Simpósio Brasileiro de Sensoriamento Remoto - SBSR**, INPE Foz do Iguaçu, PR, 2013.

ORLANDO, P.; LA ROSA, E. Object oriented methodology for change detection technique: the case of Scopello-Silicy. **South-Eastern European Journal of Earth Observation and Geomatics.** Aristotle University of Thessaloniki, Greece. v. 3, n. 2S, 2014.

RUFINO, I. A. A.; SILVA, S. T. Análise das relações entre dinâmica populacional, clima e vetores de mudança no semiárido brasileiro: uma abordagem metodológica. **Boletim de Ciências Geodésicas**, sec. Artigos, Curitiba, v. 23, n. 1, 2017, p. 166-181.

SHARMA, R.; GHOSH, A.; JOSHI, P. K. Decision tree approach for classification of remotely sensed satellite data using open source support. **J. Earth Syst. Sci**. v. 122, n. 5, p. 1237-1247. Nova Deli: Department of Natural Resources, TERI University, 2014.

SOUSA, G. M.; FERNANDES, M. C.; COSTA, G. A. O. P. Application of data mining and GEOBIA techniques for fire susceptibility analysis in the Itatiaia National Park, Brazil. **South-Eastern European Journal of Earth Observation and Geomatics**. Grécia: Aristotle University of Thessaloniki, 2014.

TSAI, C.; LINB, W.; KEC, S. Big data mining with parallel computing: A comparison of distributed and MapReduce methodologies. **The Journal of Systems and Software**, v. 122, 2016, p. 83-92.

8 – APÊNDICES

8.1 COMPROVANTE DE SUBMISSÃO DO ARTIGO 1

Boletim de Ciências Geodésicas

Preview

From: bcg_editor@ufpr.br, bcg_asseditor@ufpr.br

To: rodrigorantunes@hotmail.com

CC: rodrigorantunes@hotmail.com, edbias@unb.br, gilson.costa@ime.uerj.br, brites@unb.br

Subject: Boletim de Ciências Geodésicas - Manuscript ID BCG-2016-0058.R5

Body: 19-Sep-2017

Dear Ms. Antunes:

Your manuscript entitled "OBJECT-BASED ANALYSIS FOR URBAN SOIL MAPPING USING THE INTERIMAGE AND THE SIPINA FREE SOFTWARE PACKAGES" has been successfully submitted online and is presently being given full consideration for publication in the Boletim de Ciências Geodésicas.

Your manuscript ID is BCG-2016-0058.R5.

Please mention the above manuscript ID in all future correspondence or when calling the office for questions. If there are any changes in your street address or e-mail address, please log in to ScholarOne Manuscripts at https://mc04.manuscriptcentral.com/bcg-scielo and edit your user information as appropriate.

You can also view the status of your manuscript at any time by checking your Author Center after logging in to https://mc04.manuscriptcentral.com/bcg-scielo.

Thank you for submitting your manuscript to the Boletim de Ciências Geodésicas.

Sincerely,
Boletim de Ciências Geodésicas Editorial Office

Date Sent: 19-Sep-2017

8.2 COMPROVANTE DE SUBMISSÃO DO ARTIGO 2

REVISTA BRASILEIRA

DE CARTOGRAFIA

CAPA SOBRE PÁGINA DO USUÁRIO PESQUISA ATUAL
ANTERIORES NOTÍCIAS

Capa > Usuário > Autor > Submissões > #2143 > Avaliação

#2143 AVALIAÇÃO

RESUMO AVALIAÇÃO EDIÇÃO

SUBMISSÃO

Autores	Edilson de Souza Bias, Rodrigo Rodrigues Antunes, Ricardo Seixas Brites, Gilson Alexandre Ostwald Pinto Costa
Título	ANÁLISE DE SISTEMAS MINERADORES DE DADOS OPEN-SOURCE E SEUS ALGORITMOS DE CLASSIFICAÇÃO DE ÁRVORE DE DECISÃO INTEGRADOS COM O SISTEMA DE CLASSIFICAÇÃO BASEADA EM OBJETOS INTERIMAGE
Seção	Artigos
Editor	Alan Salomão Graça, Thales Körting

AVALIAÇÃO

RODADA 1

Versão para avaliação	2143-12890-1-RV.DOC 2017-11-29
Iniciado	2018-01-11
Última alteração	2018-02-25
Arquivo enviado	Nenhum(a)

DECISÃO EDITORIAL

Decisão	—
Notificar editor	Comunicação entre editor/autor Sem comentários
Versão do editor	Nenhum(a)
Versão do autor	Nenhum(a)
Transferir Versão do Autor	Selecionar Arquivo nenhum arquivo selecionado Transferir

IDIOMA
Português (Brasil) Submeter

USUÁRIO
Logado como: edbias
Perfil
Sair do sistema

CONTEÚDO DA REVISTA
Pesquisa
Escopo da Busca: Todos
Pesquisar
Procurar: Por Edição, Por Autor, Por Título

AUTOR
Submissões: Ativas (2), Arquivo (0), Nova submissão

TAMANHO DE FONTE
Ajuda do sistema

INFORMAÇÕES
Para leitores, Para Autores, Para Bibliotecários

8.3 COMPROVANTE DE SUBMISSÃO DO ARTIGO 3

GIScience Remote Sensing

Proof of Concept of a Novel Cloud Computing Approach for Object-Based Remote Sensing Data Analysis and Classification

--Manuscript Draft--

Full Title:	Proof of Concept of a Novel Cloud Computing Approach for Object-Based Remote Sensing Data Analysis and Classification
Manuscript Number:	TGRS-2018-0074
Article Type:	Original Article
Keywords:	Object-Based Image Analysis, InterCloud, Machine Learning
Manuscript Classifications:	Data Mining; Geocomputation; Image Classification; Information Visualization; Land Cover and Land Use
Abstract:	Advances in the development of Earth observation data acquisition systems have led to the continuously growing production of remote sensing datasets, for which timely analysis has become a major challenge. In this context, distributed computing technology can provide support for efficiently handling large amounts of data. Moreover, the use of distributed computing techniques, once restricted by the availability of physical computer clusters, is currently widespread due to the increasing offer of cloud computing infrastructure services. In this work, we introduce a cloud computing approach for object-based image analysis and classification of arbitrarily large remote sensing datasets. The approach enables exploiting machine learning methods in the creation of classification models, through the use of a web-based notebook system. A prototype of the proposed approach was implemented with the methods available in the InterCloud system integrated with the Apache Zeppelin notebook system, for collaborative data analysis and visualization. In this implementation, the Apache Zeppelin system provided the means for using the scikit-learn Python machine learning library in the creation of a classification model. In this work we also evaluated the approach with an object-based image land-cover classification of a GeoEye-1 scene, using resources from a commercial cloud computing infrastructure service provided. The obtained results showed the effectiveness of the approach in efficiently handling a large data volume in a scalable way, in terms of the number of allocated computing resources.
Order of Authors:	Rodrigo Antunes
	Thomas Blaschke
	Dirk Tiede
	Edilson Bias
	Gilson Costa
	Patrick Happ