ACA Multiagent System for Satellite Image Classification

M. Espínola[1], J.A. Piedra[1], R. Ayala[1], L. Iribarne[1], S. Leguizamón[2] and M.Menenti[3]
[1] University of Almeria,
[2] National Technological University of Argentina and
[3] Aerospace Engineering Optical and Laser Remote Sensing TUDelf

Contents

1. Motivation
2. Remote Sensing
3. Cellular Automata
4. Classification with ACA
5. Results
6. Future works

MOTIVATION

GOALS

The satellite image classification is one of the most important techniques used in remote sensing that helps on interpreting spectral bands.

Main Goal:

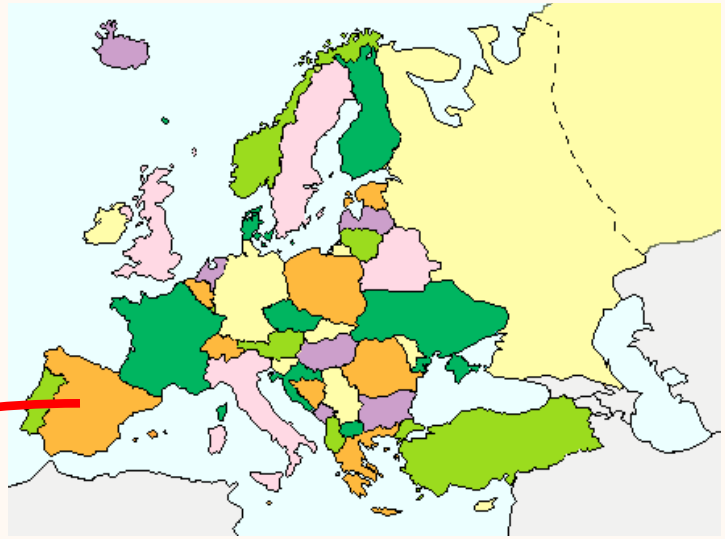
Improve the classification and the knowledge extraction through algorithms based on Cellular Automata in satellite images.

Specific goals:

1. Increase the classification quality with CA contextual techniques.
2. Define an hierarchical classification based on quality levels.
3. Obtain a list of uncertain, noisy and boundary pixels.

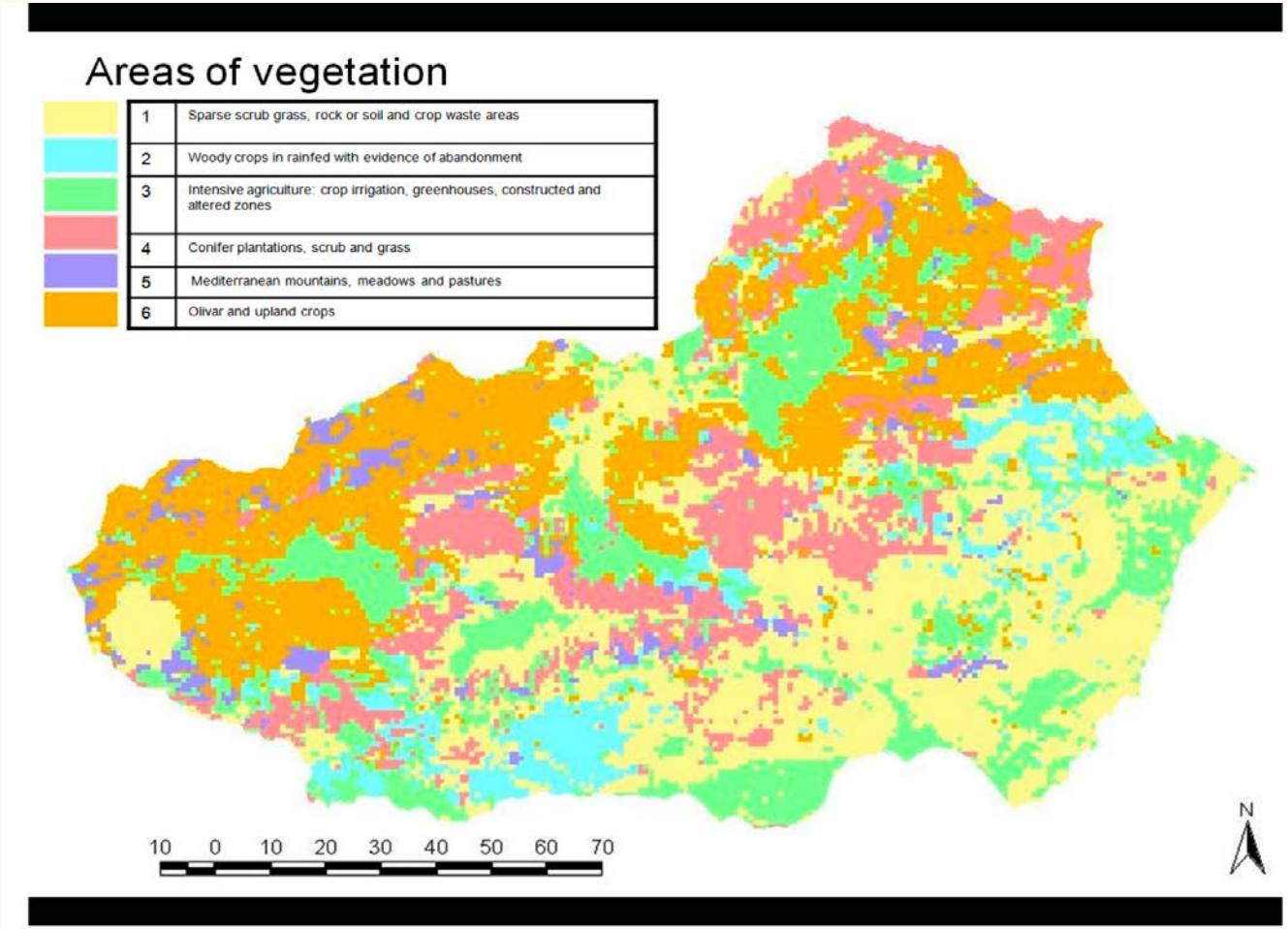
REMOTE SENSING

STUDY AREA



REMOTE SENSING

AREAS OF VEGETATION



REMOTE SENSING

SATELLITE IMAGE CLASSIFICATION ALGORITHMS

Unsupervised classification

- **K-means**
- **Isodata**

Supervised classification

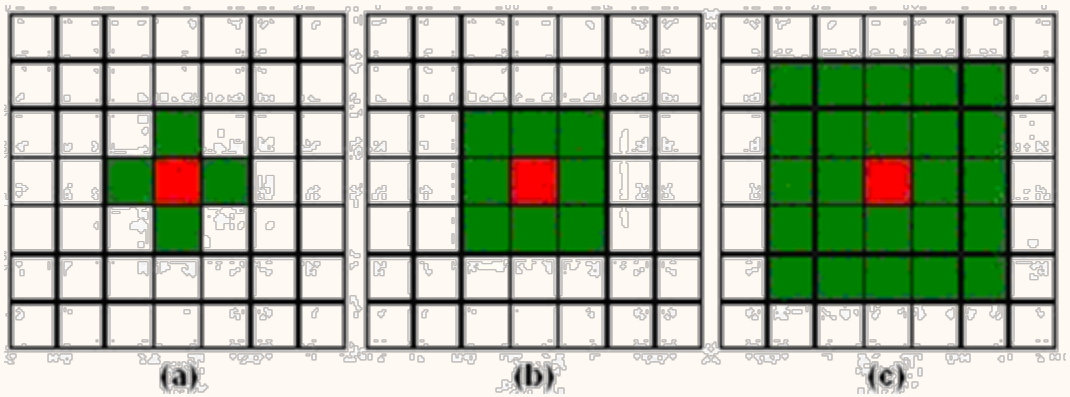
- **Minimum distance**
- **Paralelepiped**
- **Maximum likelihood**

CELLULAR AUTOMATA

DEFINITION

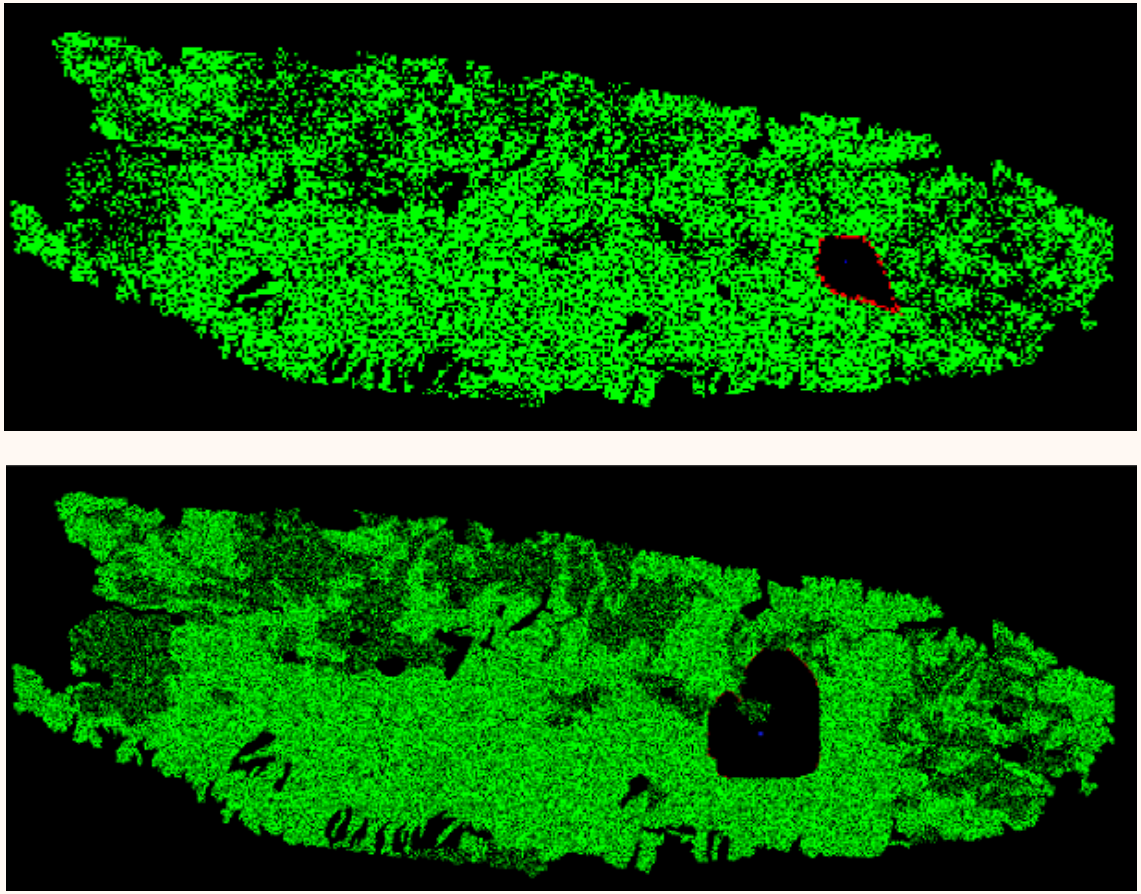
- Cellular automata: mathematical model used to study the behaviour of complex systems (von Neumann, 40s).

- Elements: cells, states, neighborhood, rules and transition function.



CELLULAR AUTOMATA

FIRE SPREAD SIMULATION



CLASSIFICATION WITH CA

INITIAL ACA CONSIDERATIONS

In order to implement ACA we must take into account the following correspondences between a cellular automaton and the basic elements of a generic process of satellite image classification:

- Each cell of the grid corresponds to a pixel of the image.
- Each state of cellular automaton will represent a different class of the final classification.
- The neighbourhood of each cell will consist of the 8 nearest cells (Moore neighbourhood).
- The transition function *f* must correctly classify each pixel of the image based on the features of the current cell and its neighbourhood, using mixed spectral and contextual data.

CLASSIFICATION WITH CA

ACA STATES

We have assigned 3 states for each cell: [class][quality][type], where each state can take the following values:

- [class]=training set classes, noiseClass (noisy pixels) or emptyClass (pixels not classified yet).
- [quality]= 1..numIterations (number of iterations by CA)
- [type]= focus (not boundary pixels), boundary (pixels from edge), uncertain (caotic pixels) and noisy (noise detection).

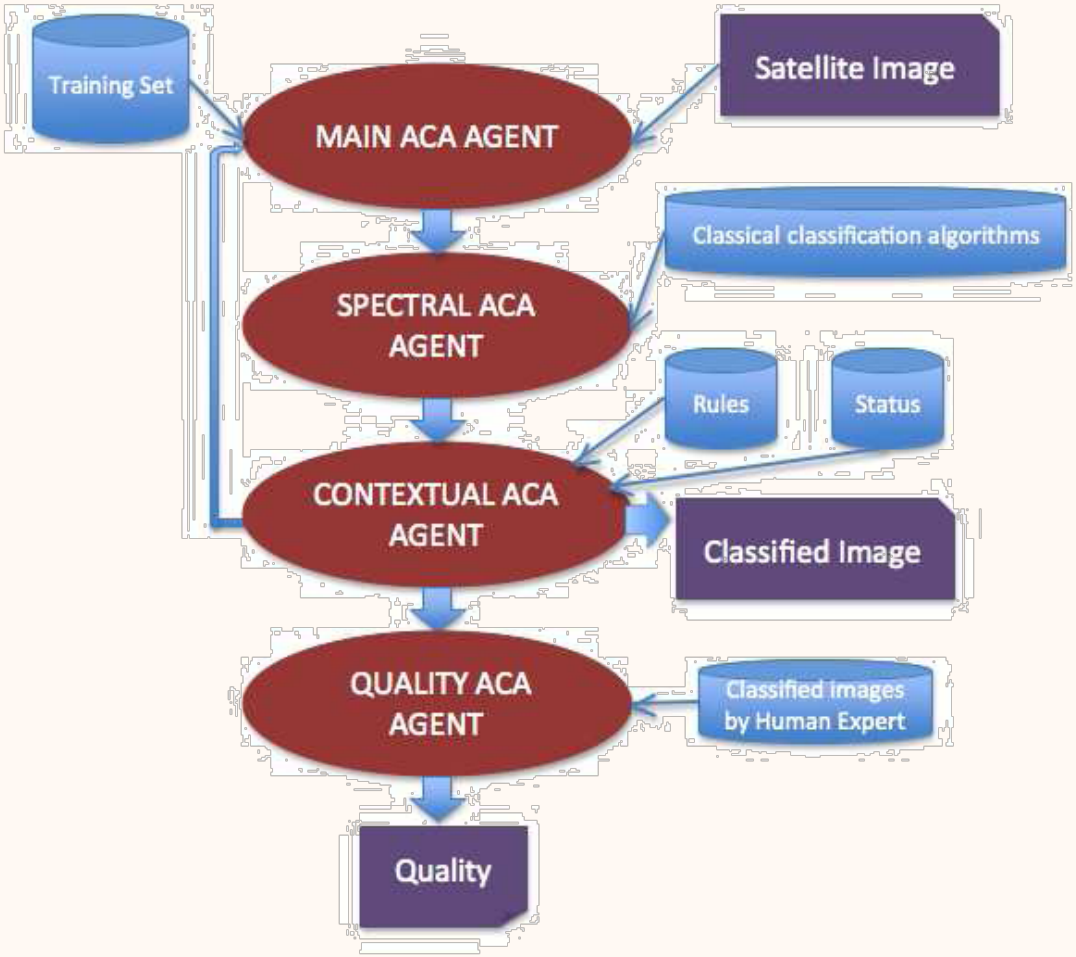
CLASSIFICATION WITH CA

ACA RULES

- Rule 1. If the number of spectral classification classes is 1, and the neighbourhood class states are emptyClass or the same as actual pixel: [class][quality][type] = spectralClass, CAiteration, *focus*
- Rule 2. If the number of spectral classification classes is 1, and the neighbourhood class states are different than actual pixel class: [class][quality][type] = spectralClass, CAiteration, *boundary*
- Rule 3. If the number of spectral classification classes is 1 and the spectralClass is noiseClass: [class][quality][type] = majority class of the neighbourhood, CAiteration, *noisy*
- Rule 4. If the number of spectral classification classes is bigger than 1: [class][quality][type] = majority class of the neighbourhood among the dubious classes, CAiteration, *uncertain*

CLASSIFICATION WITH CA

ACA MULTIAGENT ARCHITECTURE



CLASSIFICATION WITH CA

MAIN ACA AGENT PSEUDOCODE

Main ACA algorithm $(E, numClasses, numIterations, threshold)$

Input: $E = \{e_1, e_2, ..., e_n\}$: set of pixels to classify
$numClasses$: number of classes
$numIterations$: CA maximum iterations
$threshold$: threshold for class membership

Output: $C = \{c_1, c_2, ..., c_k\}$: set of class centers
$L = \{l(e) | e = 1, 2, ..., n\}$: classified pixels

```
01 for i ← 0 to numIterations do
02   foreach e_i ∈ E do
03     if e_i.classificated ≠ true then
04       spectralClasses ← spectralACA(e_i, numClasses, threshold);
05       if spectralClasses ≠ ∅ then
06         finalClass ← contextualACA(e_i, spectralClasses);
07       endif
08     endif
09   end
10   threshold ← threshold + incremental;
11 end
```

CLASSIFICATION WITH CA

SPECTRAL ACA AGENT PSEUDOCODE

Spectral ACA algorithm $(e_i, numClasses, threshold)$

Input: e_i: pixel to classify
$numClasses$: number of classes
$threshold$: threshold for class membership

Output: $spectralClasses$: classes which may belong the pixel

```
01 spectralClasses ← ∅;
02 for j ← 0 to numClasses do
03    if minDistance(e_i, class_j) | j ∈ {1..k} ≤ threshold then
04       spectralClasses ← spectralClassesUclass_j;
05    endif
06 end
```

CLASSIFICATION WITH CA

CONTEXTUAL ACA AGENT PSEUDOCODE

Contextual ACA algorithm $(e_i, spectralClasses)$

Input:

e_i: pixel to classify

$spectralClasses$: classes which may belong the pixel

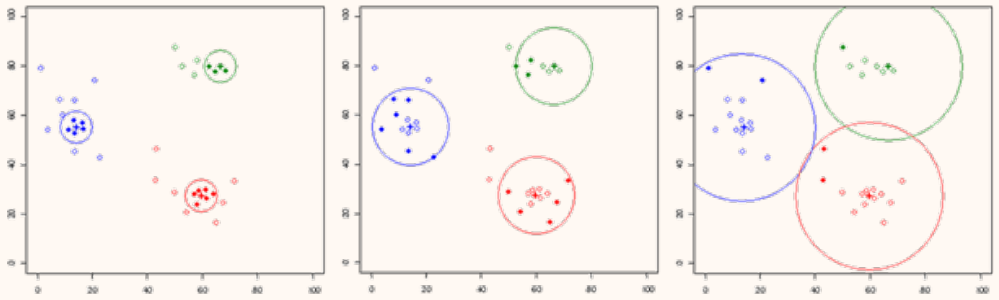
Output:

$finalClass$: final class of the pixel

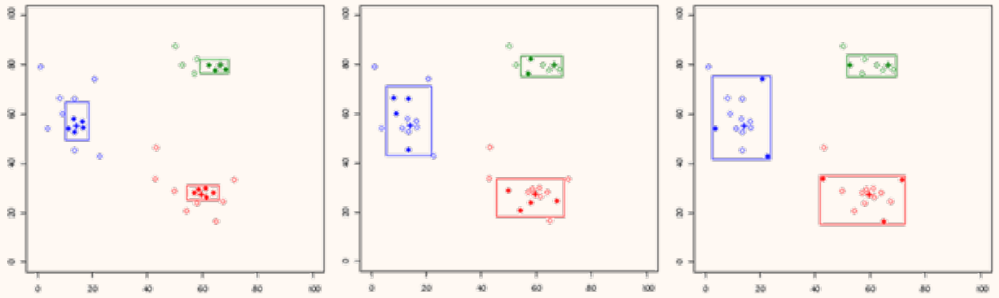
```
01  if size(spectralClasses) = 1 then
02    if{spectralClasses} ≠ noiseClass then
03      e_i.class ← {spectralClasses};
04      e_i.quality ← numIteration;
05      if neighbourhoodClassesType(e_i) = 1 then
06        e_i.type ← focusPixel;
07      endif
08      if neighbourhoodClassesType(e_i) = 2 then
09        e_i.type ← edgePixel;
10      endif
11    endif
12    if{spectralClasses} = noiseClass then
13      e_i.class ← bayesNeighbourhood();
14      e_i.quality ← numIteration;
15      e_i.type ← noisePixel;
16    endif
17  endif
18  if size(spectralClasses) ≠ 1 then
19    e_i.class ← bayesNeighbourhoodClass();
20    e_i.quality ← numIteration;
21    e_i.type ← uncertainPixel;
22  endif
```

CLASSIFICATION WITH CA

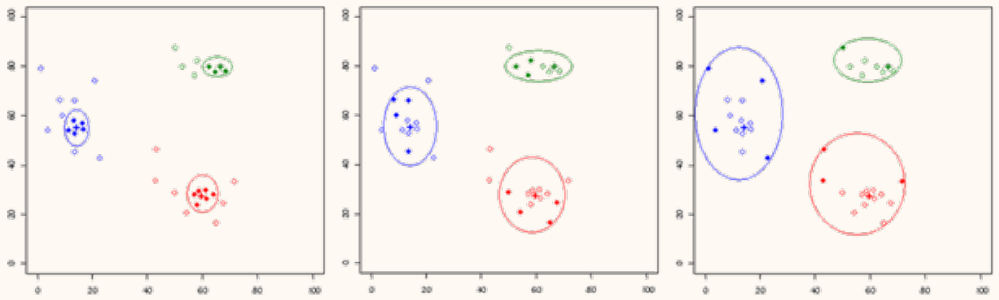
ACA CLASSIFICATION PROCESS SIMULATION



Minimum distance



Paralelepiped

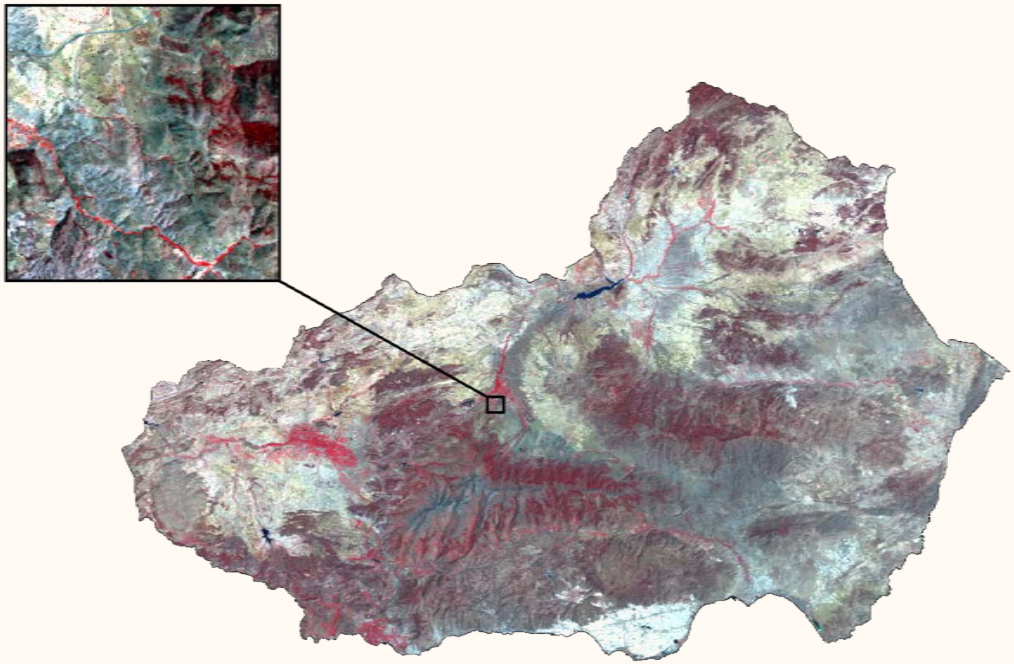


Maximum likelihood

RESULTS

SATELLITE IMAGE USED

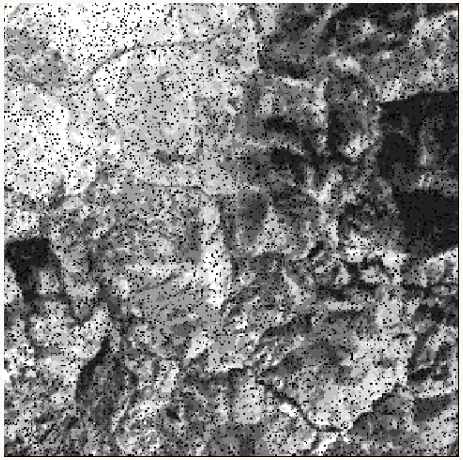
Tests have been carried out on a multispectral Landsat image with 7 layers, where the spatial resolution of each pixel is 30x30 meters.



RESULTS

GOAL 1. IMPROVE THE CLASSIFICATION QUALITY (CONTEXTUAL)

Expert field classification



Minimum distance classification

	Class 1	Class 2	Class 3	Class 4	Class 5	Class 6	Class 7	Class 8
Class 1	**0**	0	0	0	0	0	0	0
Class 2	789	**6080**	355	0	0	0	0	0
Class 3	1247	0	**9447**	532	0	0	0	0
Class 4	1547	0	2	**11998**	242	0	0	0
Class 5	1555	0	0	52	**12827**	3	27	0
Class 6	1027	0	0	47	281	**8330**	35	1
Class 7	1513	0	0	0	250	12	**13050**	0
Class 8	1396	0	0	0	0	66	381	**11242**

ACA minimum distance confusion matrix

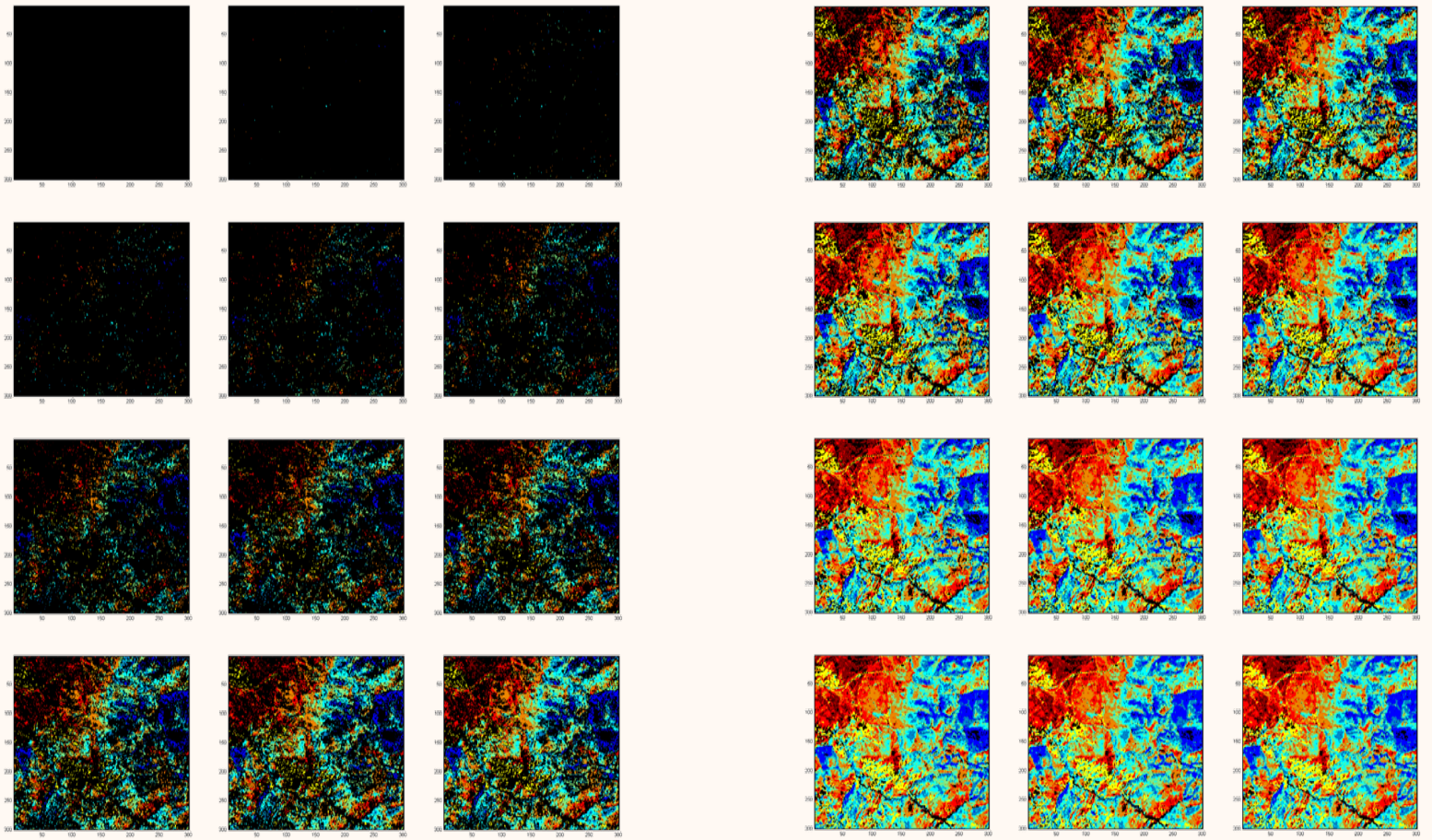
Correct classified pixels with classical minimum distance: 85%
Correct classified pixels with ACA minimum distance: 90%

Correct classified pixels with classical paralelepiped: 89%
Correct classified pixels with ACA paralelepiped: 92%

Correct classified pixels with classical maximum likelihood: 90%
Correct classified pixels with ACA maximum likelihood: 94%

RESULTS

GOAL 2. HIERARCHICAL CLASSIFICATION (QUALITY LEVELS)



Hierarchical classification in 100 iterations of ACA algorithm

RESULTS

GOAL 3. LIST OF UNCERTAIN, NOISY AND BOUNDARY PIXELS

Pixel	Clases	Pixel	Clases	Pixel	Clases	Pixel	Clases
[2, 218]	5, 7	[93, 92]	7, 8	[174, 32]	4, 5	[238, 259]	5, 7
[3, 161]	5, 7	[94, 109]	7, 8	[174, 253]	7, 8	[239, 34]	7, 8
[3, 162]	5, 7	[94, 248]	4, 5	[175, 159]	3, 5	[242, 173]	6, 3
[3, 293]	4, 5	[94, 249]	4, 5	[175, 169]	4, 5	[243, 206]	7, 8
[6, 241]	1, 2	[95, 296]	1, 2	[178, 146]	8, 6	[246, 97]	7, 8
[7, 9]	5, 3	[96, 123]	7, 8	[179, 32]	4, 5	[247, 259]	7, 8
[7, 245]	1, 2	[97, 282]	4, 5	[182, 21]	2, 4	[248, 288]	1, 2
[8, 206]	2, 4	[99, 146]	5, 7	[182, 169]	4, 5	[250, 82]	5, 7
[8, 207]	2, 4	[103, 108]	4, 5	[185, 30]	7, 8	[251, 117]	4, 5
[8, 271]	2, 4	[104, 108]	4, 5	[186, 187]	4, 5	[251, 136]	2, 4
[10, 190]	2, 4	[104, 134]	7, 8	[186, 267]	4, 5	[252, 157]	5, 7
[11, 164]	4, 5	[104, 276]	5, 7	[189, 300]	4, 5	[253, 134]	2, 4
[12, 168]	5, 7	[115, 270]	2, 4	[190, 199]	5, 7	[254, 116]	5, 7
[14, 294]	4, 5	[116, 191]	4, 5	[194, 50]	5, 7	[255, 161]	7, 8
[18, 144]	5, 7	[117, 191]	4, 5	[195, 274]	2, 4	[256, 81]	8, 6
[20, 169]	4, 5	[118, 35]	8, 6	[195, 291]	2, 4	[257, 81]	8, 6
[21, 169]	4, 5	[118, 194]	2, 4	[196, 232]	5, 7	[258, 300]	2, 4
[26, 145]	7, 8	[118, 260]	2, 4	[196, 291]	2, 4	[260, 107]	7, 3
[27, 251]	4, 5	[119, 260]	2, 4	[197, 57]	5, 3	[260, 241]	7, 8
[28, 285]	5, 7	[123, 275]	3, 5	[197, 73]	4, 5	[260, 242]	7, 8
[29, 174]	4, 5	[126, 38]	4, 5	[197, 300]	1, 2	[262, 71]	2, 4
[34, 230]	2, 4	[126, 282]	1, 2	[198, 290]	4, 5	[262, 205]	5, 7
[34, 250]	5, 7	[127, 239]	5, 7	[198, 300]	1, 2	[263, 17]	4, 5
[36, 213]	2, 4	[127, 240]	5, 7	[199, 16]	2, 4	[263, 48]	2, 4
[38, 206]	5, 7	[133, 178]	4, 5	[199, 83]	5, 7	[264, 70]	2, 4
[43, 236]	4, 5	[133, 225]	5, 7	[199, 198]	5, 7	[270, 210]	7, 8
[43, 260]	4, 5	[137, 156]	4, 5	[200, 22]	4, 5	[271, 55]	4, 5
[44, 286]	5, 7	[141, 90]	8, 6	[200, 26]	4, 5	[274, 121]	2, 4
[45, 278]	5, 7	[141, 174]	5, 7	[203, 76]	2, 4	[278, 166]	8, 6
[50, 230]	2, 4	[142, 174]	5, 7	[203, 238]	4, 5	[279, 279]	5, 7
[51, 243]	5, 7	[144, 202]	4, 5	[204, 61]	5, 7	[280, 93]	4, 5
[52, 197]	5, 7	[145, 231]	5, 7	[205, 18]	2, 4	[281, 45]	2, 4
[55, 52]	7, 8	[148, 134]	7, 8	[205, 59]	4, 5	[282, 226]	8, 6

FUTURE WORK

IMPROVEMENTS OF ACA ALGORITHM

Some possible future work are shown below:

- Implementing new versions of the ACA algorithm based on **new states and rules** of cellular automaton to further **customize the classification process**.
- **Creating a ERDAS Imagine pluggin** that allows a custom classification based on cellular automata.

THANK YOU FOR THE ATTENTION

ACA Multiagent System for Satellite Image Classification

M. Espínola[1], J.A. Piedra[1], R. Ayala[1], L. Iribarne[1], S. Leguizamón[2] and M.Menenti[3]
[1] University of Almeria,
[2] National Technological University of Argentina and
[3] Aerospace Engineering Optical and Laser Remote Sensing TUDelf