



# PROYECTO FINAL

PROCESAMIENTO DE IMAGENES DEEP LEARNING  
MAESTRIA EN ESTADISTICA APLICADA

JHON FREDYS BELLO CORDERO  
JORGE LUIS COHEN TORRES

1

# SEGMENTACION DE ARBORIO

Imagen original RGB



Imagen en blanco y negro



1

## SEGMENTACION DE BASMITI

Imagen original RGB



Imagen en blanco y negro



1

# SEGMENTACION DE IPSALA

Imagen original RGB



Imagen en blanco y negro



1

## SEGMENTACION DE JASMINE

Imagen original RGB



Imagen en blanco y negro



1

## SEGMENTACION DE KARACADAG

Imagen original RGB



Imagen en blanco y negro



1

# MORFOLOGIA KARACADAG

Dilataci n



Erosi n



Apertura

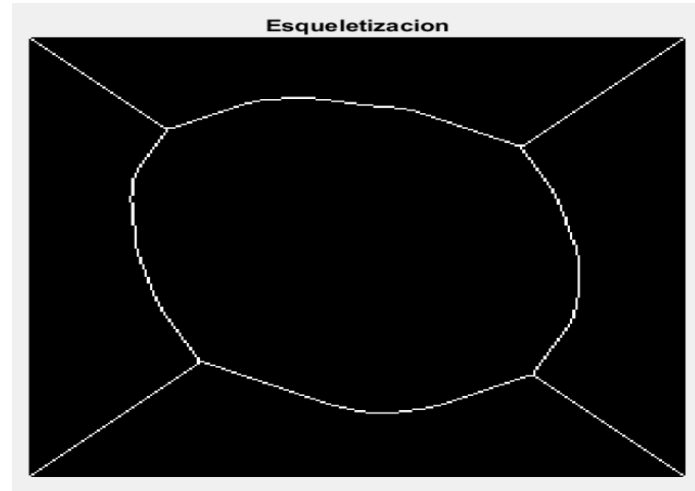


Cierre



1

# MORFOLOGIA KARACADAG





1

# MORFOLOGIA IPSALA

Dilatación



Erosión



Apertura



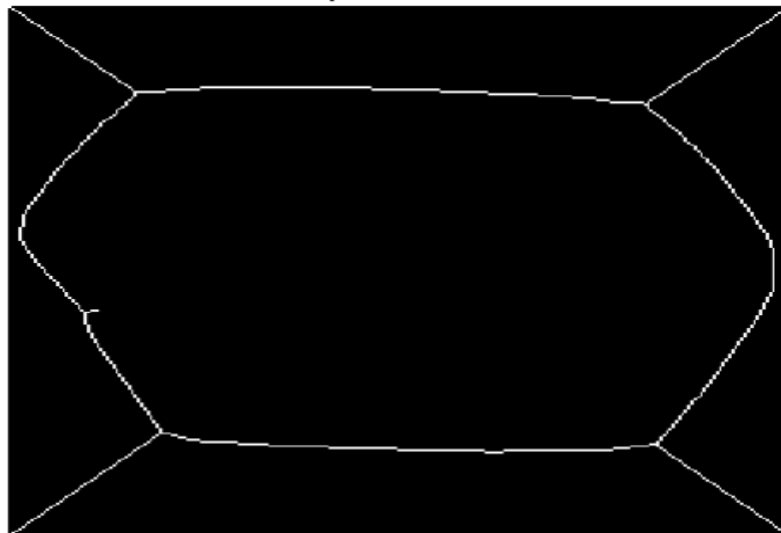
Cierre



1

# MORFOLOGIA IPSALA

Esqueletización



# RANDOM FOREST CON 2 CARACTERISTICAS... (88%)

```
54 end
55
56 %% Configuración y ejecución del clasificador
57 % 1.1 Configuración input
58 %X = [area; perim; orien; circ; ejeMen; ejeMay]'; %Datos de entrenamiento
59 %Y = [area; circ]'; %Datos de entrenamiento
60
61 % 1.2 Configuración target
62 T = repmat([1 2 3 4 5],5,1); %Se crea una matriz con las etiquetas posibles para las muestras (2 filas, 1 columna)
63 target = T(:); %Se serializa T, Vector objetivo para la clasif supervisada
64 %target = target';
65
66 %% Configuración y obtención del RF
67 disp('Configuring Random Forest...');
68 nArboles = 50; % número de árboles
69 myRF = TreeBagger(nArboles, X, target, 'OOBPrediction', 'on');
70
71 %% Validación del modelo sobre muestras de prueba (las que no se usaron durante el entrenamiento)
72 % Se toman las 2 segundas muestras de cada clase
73 j=1;
74 for i = 6:10:ss(1) %Cada 4 posiciones del vector se toman las 2 segundas muestras de cada clase
75     area(j) = matr_descrip(i,1);
```

Command Window

How to MATLAB? See resources for Getting Started.

outputs =

1	1	5	1	1	2	4	2	2	2	3	3	3	3	3	4	4	4	4	4	5	5	5	5	1	
ins =	1	1	1	1	1	2	2	2	2	2	3	3	3	3	3	4	4	4	4	4	5	5	5	5	5
val =	88																								



# RANDOM FOREST CON 4 CARACTERISTICAS... (92%)

```
55
56 %% Configuración y ejecución del clasificador
57 % 1.1 Configuración input
58 %% X = [area; perim; orien; circ; ejeMen; ejeMay]'; %Datos de entrenamiento
X = [area; circ; MajorAxisLength; MinorAxisLength]'; %Datos de entrenamiento
61 % 1.2 Configuración target
62 T = repmat([1 2 3 4 5],5,1); %Se crea una matriz con las etiquetas posibles para las muestras (2 filas, 1 columna)
63 target = T(:); %Se serializa T, Vector objetivo para la clasif supervisada
64 %target = target';
65
66 %% Configuración y obtención del RF
67 disp('Configuring Random Forest...');
68 nArboles = 50; % número de árboles
69 myRF = TreeBagger(nArboles, X, target, 'OOBPrediction','on');
70
71 %% Validación del modelo sobre muestras de prueba (las que no se usaron durante el entrenamiento)
72 % Se toman las 2 segundas muestras de cada clase
73 j=1;
```

Command Window

How to MATLAB? See resources for Getting Started.

Outputs =

1	1	1	1	1	2	2	2	2	2	3	3	3	3	3	4	4	4	4	4	4	5	5	5	5	5
1	1	1	1	1	2	2	2	2	2	3	3	3	3	3	4	4	4	4	4	4	5	5	5	5	5
1	1	1	1	1	2	2	2	2	2	3	3	3	3	3	4	4	4	4	4	4	5	5	5	5	5

Accuracy =

92



# CONFUSIONES 1

X ARBORIO POSICIÓN 6



X KARACADAG POSICION 6



Se están confundiendo Arborio de la posición 6 con karacadag de la posición 6

## CONFUSIONES 2

X ARBORIO POSICIÓN 9



X KARACADAG POSICIÓN 9



Se están confundiendo Arborio de la posición 9 con karacadag de la posición 9

# RED NEURONAL CON 2 CARACTERISTICAS... (56%)

```
55
56 %% Configuración y ejecución del clasificador
57 % 1.1 Configuración input
58 %X = [area; perim; orien; circ; ejeMen; ejeMay]'; %Datos de entrenamiento
59 X = [area; circ;]'; %Datos de entrenamiento
60
61 % 1.2 Configuración target
62 T = repmat([1 2 3 4 5],5,1); %Se crea una matriz con las etiquetas posibles para las muestras (2 filas, 9 columnas)
63 target = T(:); %Se serializa T, Vector objetivo para la clasif supervisada
64 %target = target';
65
66 % 2: Configuración de la red neuronal
67 disp('Configuring Neural Network...');
68 trainFcn = 'trainlm'; % Levenberg-Marquardt
69 hiddenLayerSize = [3 15 5]; %If I need more layers then I should write: [10,12,...,9]
70 net = fitnet(hiddenLayerSize,trainFcn);
71 % net.layers{1}.transferFcn='logsig'; %tansig by default, but I can put another
72 % net.layers{2}.transferFcn='tansig';
73 % net.layers{3}.transferFcn='purelin';
74 % net.trainParam.goal = 0.1;
75 % net.trainParam.epochs = 500;
76 %net = init(net); %initializing the network with previous configurations
```

Command Window

How to MATLAB? See resources for Getting Started.

Input:

0	4	4	5	1	2	2	2	2	2	3	3	3	3	3	4	5	5	5	4	4	4	4	4	5
1	1	1	1	1	2	2	2	2	2	3	3	3	3	3	4	4	4	4	4	5	5	5	5	5

Final =

56.0000



# RED NEURONAL CON 4 CARACTERISTICAS... (80%)

```
55
56 %% Configuración y ejecución del clasificador
57 % 1.1 Configuración input
58 %X = [area; perim; orien; circ; ejeMen; ejeMay]'; %Datos de entrenamiento
59 X = [area; circ; MajorAxisLength; MinorAxisLength]'; %Datos de entrenamiento
60
61 % 1.2 Configuración target
62 T = repmat([1 2 3 4 5],5,1); %Se crea una matriz con las etiquetas posibles para las muestras (2 filas, 9 columnas)
63 target = T(:); %Se serializa T, Vector objetivo para la clasif supervisada
64 %target = target';
65
66 % 2: Configuración de la red neuronal
67 disp('Configuring Neural Network...');
68 trainFcn = 'trainlm'; % Levenberg-Marquardt
69 hiddenLayerSize = [3 15 5]; %if I need more layers then I should write: [10,12,...,9]
70 net = fitnet(hiddenLayerSize,trainFcn);
71 % net.layers{1}.transferFcn='logsig'; %tansig by default, but I can put another
72 % net.layers{2}.transferFcn='tansig';
73 % net.layers{3}.transferFcn='purelin';
74 % net.trainParam.goal = 0.1;
75 % net.trainParam.epochs = 500;
76 %net = init(net); %initializing the network with previous configurations
```

Command Window

How to MATLAB? See resources for Getting Started.

tputs =

1 1 1 1 1 2 2 3 2 2 3 3 2 3 3 4 4 4 4 4 2 5 5 1 3

s =

1 1 1 1 1 2 2 2 2 2 3 3 3 3 3 4 4 4 4 4 5 5 5 5 5

a1 =

80





# RED NEURONAL CON 4 CARACTERISTICAS Y 2 CAPAS... (72%)

```
55
56 %% Configuración y ejecución del clasificador
57 % 1.1 Configuración input
58 %X = [area; perim; orien; circ; ejeMen; ejeMay]'; %Datos de entrenamiento
59 X = [area; circ; MajorAxisLength; MinorAxisLength]'; %Datos de entrenamiento
61 % 1.2 Configuración target
62 T = repmat([1 2 3 4 5],5,1); %Se crea una matriz con las etiquetas posibles para las muestras (2 filas, 9 columnas)
63 target = T(:); %Se serializa T, Vector objetivo para la clasif supervisada
64 %target = target';
65
66 % 2: Configuración de la red neuronal
67 disp('Configuring Neural Network...');
68 trainFcn = 'trainlm'; % Levenberg-Marquardt
69 hiddenLayerSize = [10 15]; %If I need more layers then I should write: [10,12,...,9]
70 net = fitnet(hiddenLayerSize,trainFcn);
71 % net.layers{1}.transferFcn='logsig';
72 % net.layers{2}.transferFcn='tansig'; %tansig by default, but I can put another
73 % net.layers{3}.transferFcn='purelin';
74 % net.trainParam.goal = 0.1;
75 % net.trainParam.epochs = 500;
76 %net = init(net); %initializing the network with previous configurations
```

Command Window

new to MATLAB? See resources for Getting Started.

```
Inputs =
    1     0     0     1     1     2     2     2     2     2     3     3     3     3     3     4     4     4     4     4     2
    2     2     2     1     2

as =
    1     1     1     1     2     2     2     2     2     3     3     3     3     3     4     4     4     4     4     5
    5     5     5     5     5

val =
    72
```



# LINKS

<https://github.com/jcohentorres/JORGEE-COHEN---JHON-BELLO>

# MUCHAS GRACIAS

