



NEVER URUETA PEÑATA DIEGO HERRERA MALAMBO

PROCESAMIENTO DE IMÁGENES Y
APRENDIZAJE DE MAQUINA

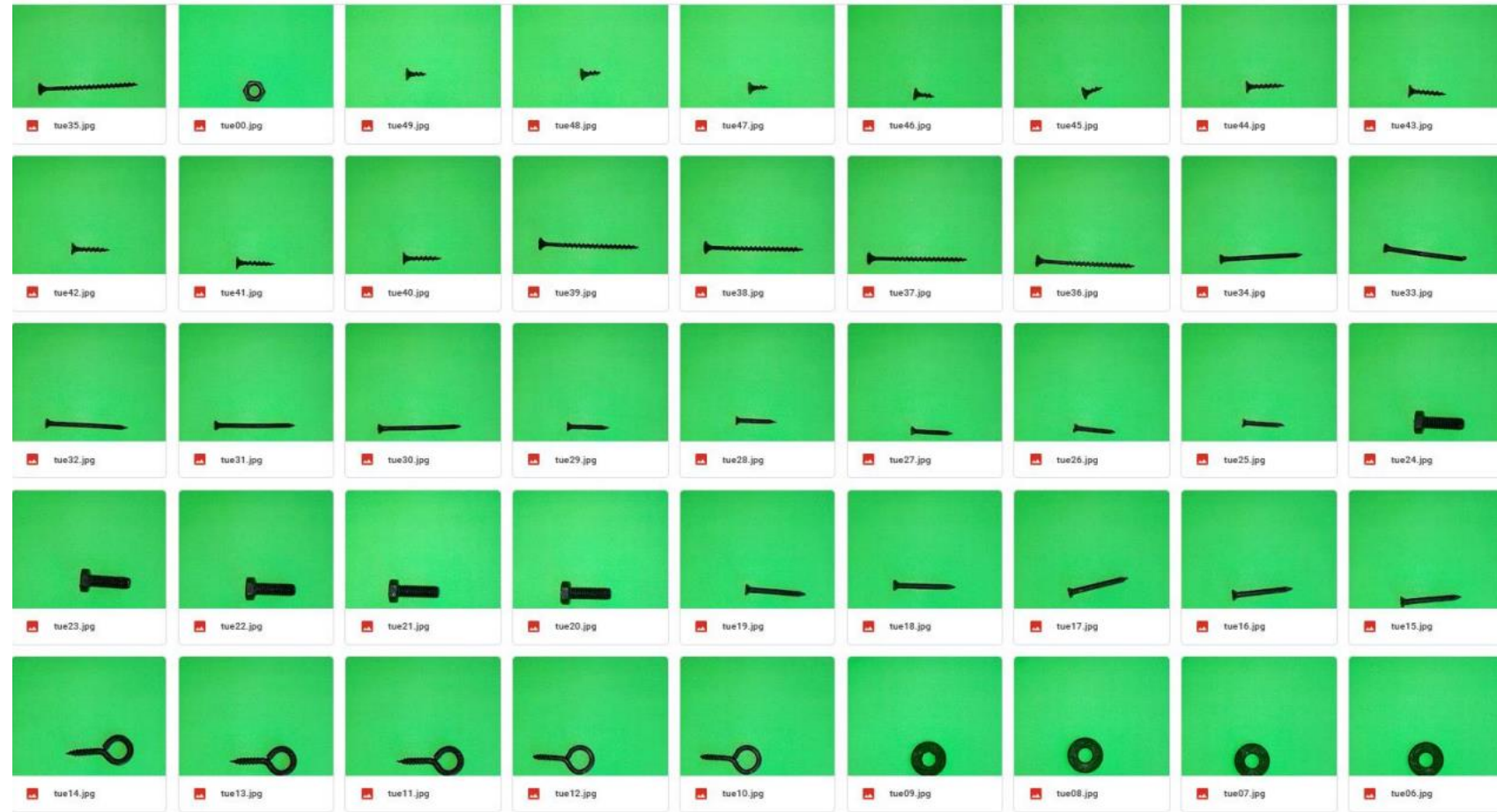
PROYECTO FINAL

MAESTRÍA EN ESTADÍSTICA APLICADA
UNIVERSIDAD TECNOLÓGICA DE BOLÍVAR

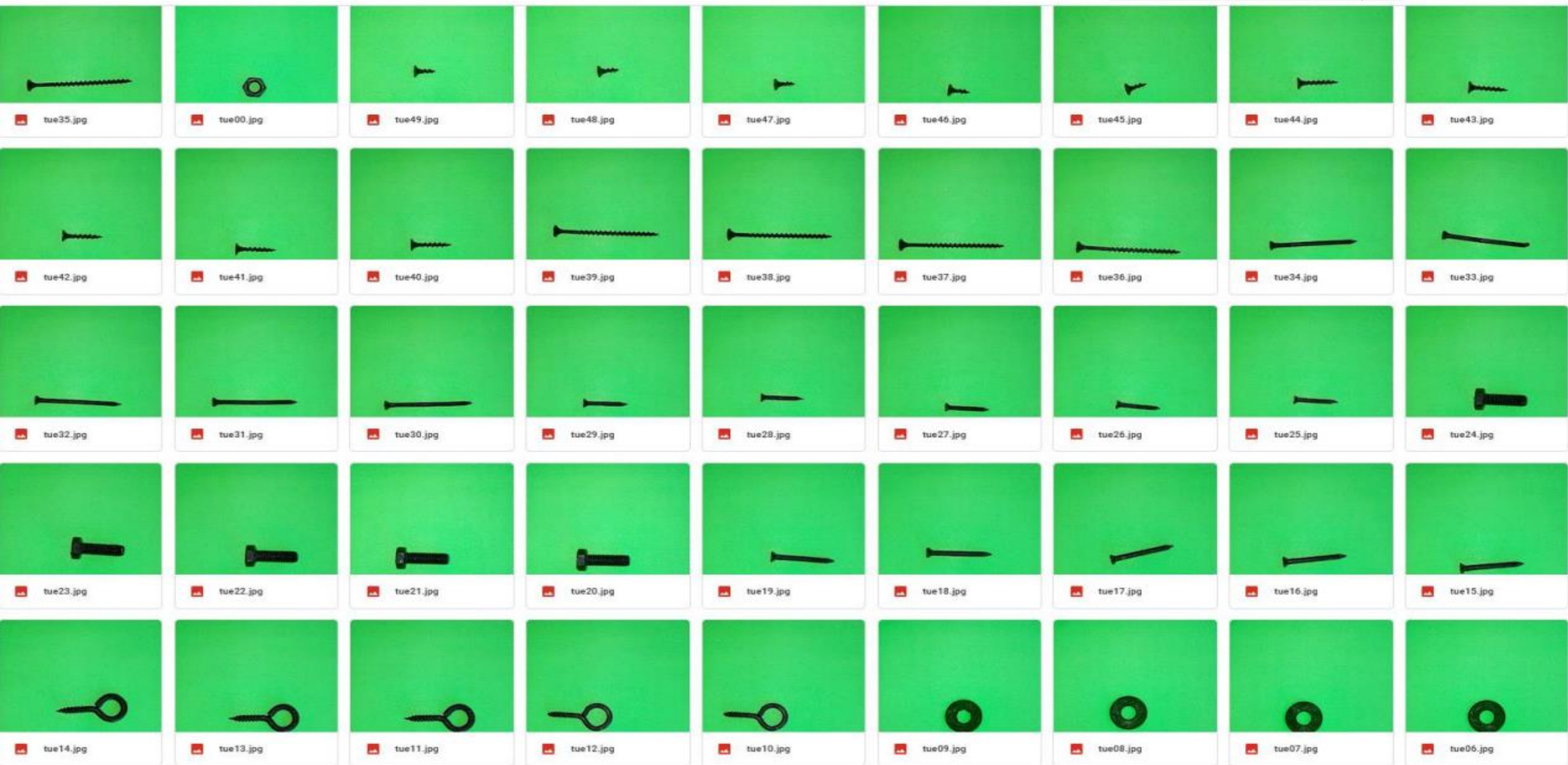
2022-2

DEFINICIÓN DEL PROBLEMA

Diferenciar los objetos de ferretería de la base de datos proporcionada, mediante herramientas de Machine Learning. La información seleccionada es aquella que se genera posterior al tratamiento de las imágenes tomadas

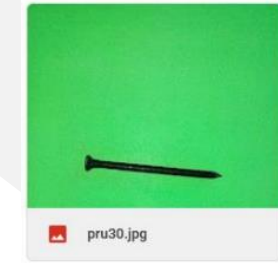


DATA ORIGINAL



CARACTERISITCAS A IDENTIFICAR

- Forma geométrica: alargada, cabeza redonda y puntiagudo al final
- Largo: distintos (pequeño, mediano, largo)
- Ancho: delgado
- Espesor: NA
- Bordes de la forma: con espirales
- Bordes de la forma: con espirales



- Forma geométrica: hexagonal
- Largo: distintos (pequeño, mediano)
- Ancho: distintos (pequeño y mediano)
- Espesor: diámetro interior y exterior.
- Bordes de la forma: NA



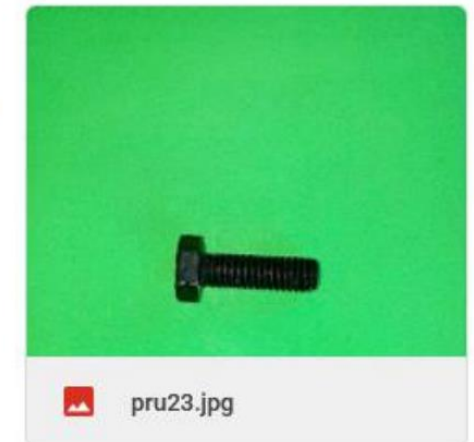
- Forma geométrica: redondo con hueco
- Largo: distintos (pequeño, mediano)
- Ancho: distintos (pequeño y mediano)
- Espesor: diámetro interior y exterior.
- Bordes de la forma: NA



- Forma geométrica: unión de forma alargada con forma redonda y puntiaguda al final
- Largo: NA
- Ancho: NA
- Espesor: NA
- Bordes de la forma: con espiral la parte alargada y lisa la parte redonda



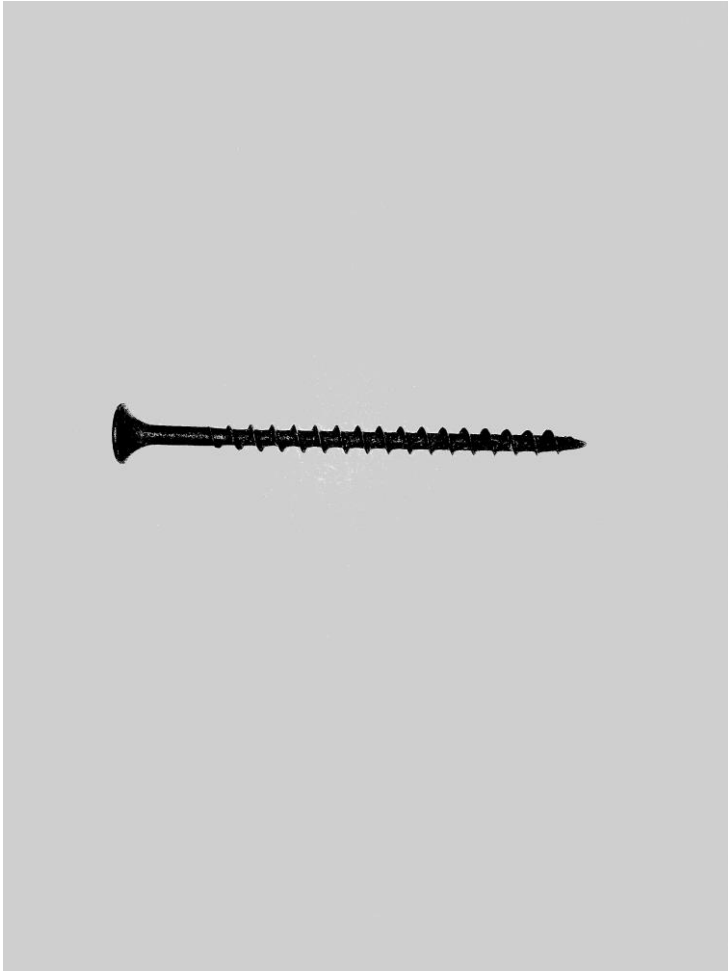
- Forma geométrica: unión de forma alargada una cabeza ancha y punta roma
- Largo: mediano
- Ancho: mediano
- Espesor: NA
- Bordes de la forma: con espiral la parte alargada



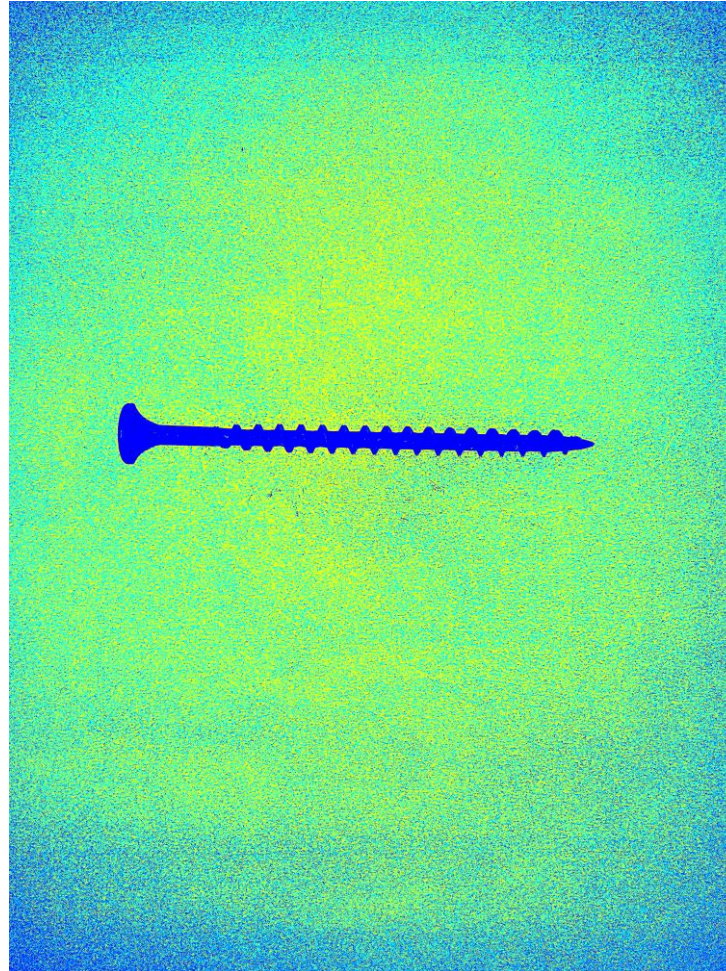
PRE-PROCESADO DE DATOS

Segmentación: Manual – 1 Umbral

Manual



Otsu



PRE-PROCESADO DE DATOS

Cierre + Erosion: square



Cierre + Erosion: ball



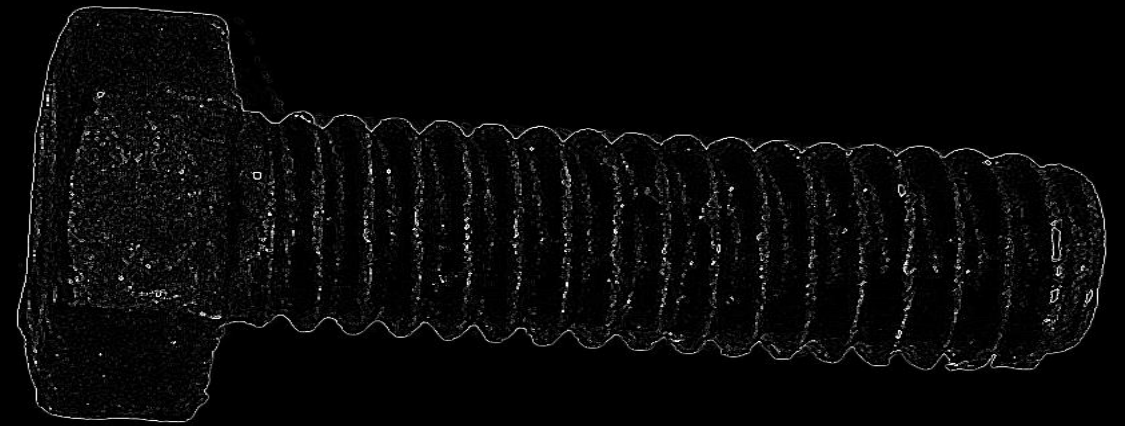
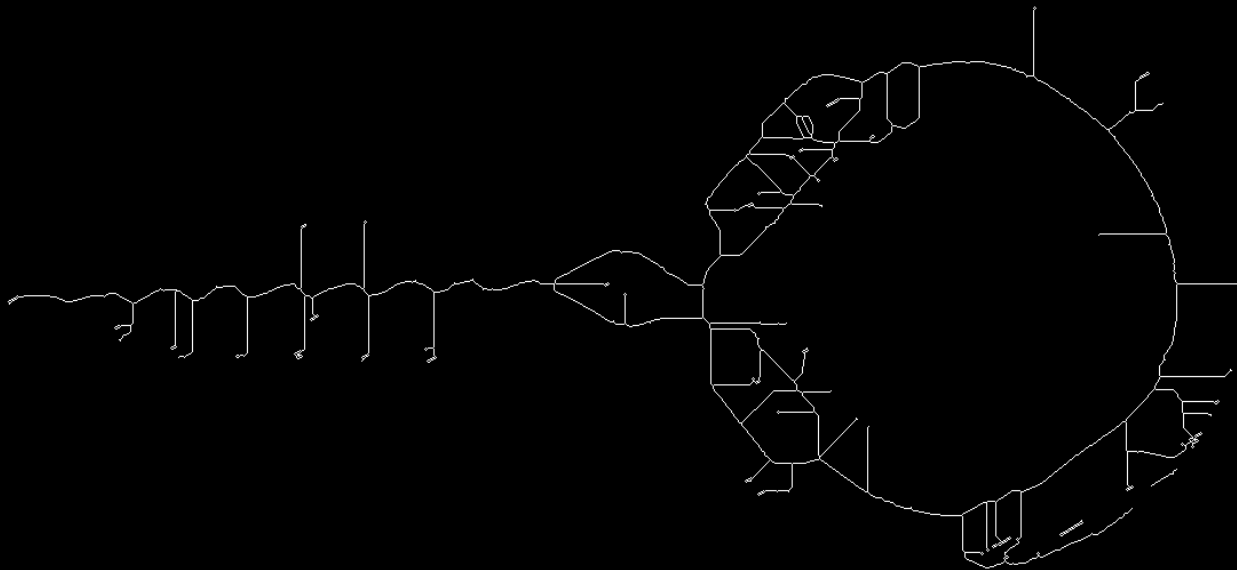
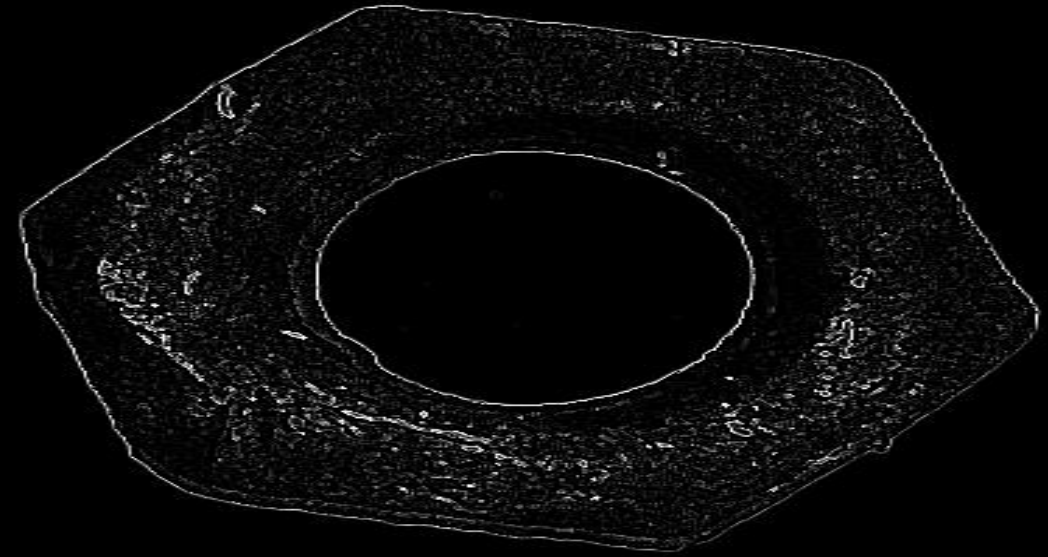
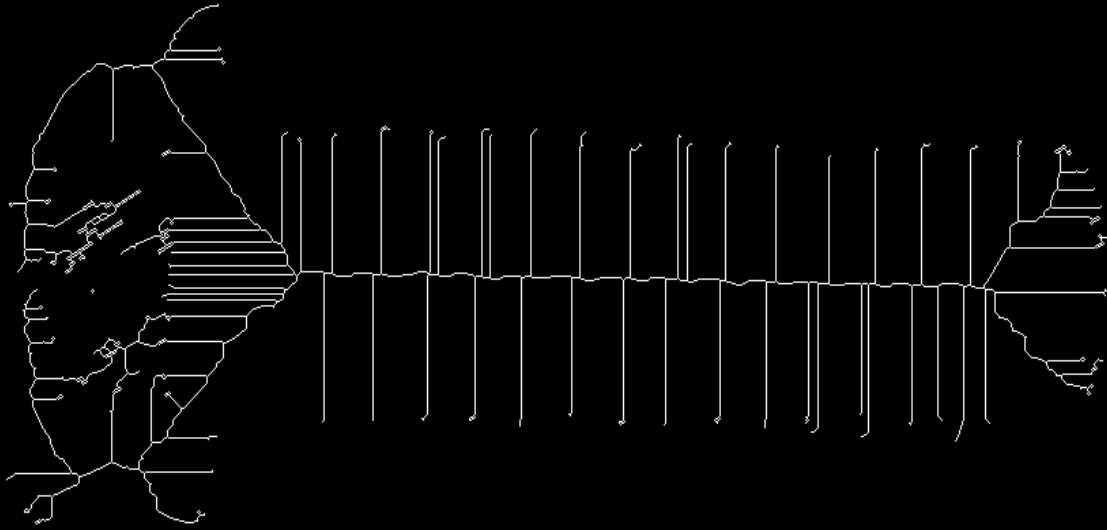
Apertura(line) + Dilatacion(line)
Llenado de huecos



PRE-PROCESADO DE DATOS

Cierre + Erosión: Square





EXTRACCIÓN DE LA INFORMACIÓN Y ANÁLISIS DE DATOS

Características: "Area", "Circularity", "FilledArea", "Perimeter", "MajorAxisLength", "Puntas" **RANDOM FOREST**

Current Folder

- Clase
- PROYECTO FINAL LISTO
 - classif_ANN_D.m
 - classif_ANNx.m
 - classif_RFx.m
- Descriptores_Prueba_Final.xlsx
- Descriptores_Prueba_sin_correccion.xlsx
- Descriptores_Prueba.xlsx
- Procesador.m

Workspace

Name	Value	Size	Class
ans	1×20 double	1×20	double
campos_a_ev...	1×6 string	1×6	string
elementos_test	2	1×1	double
eval	95	1×1	double
fName	'/MATLAB Drive/P...	1×72	char
matr_descrip	50×26 table	50×26	table
myRF	1×1 TreeBagger	1×1	TreeBagger
nArboles	100	1×1	double
numero_elem...	5	1×1	double
outputs	1×20 double	1×20	double

Tarea_5_Morfologia.m x Tarea_6_Reconocimiento_Patrones.m x Tarea_5.m x S3_Reconocimiento_patrones.m x Procesador.m x classif_RFx.m x +

```
13 nArboles = 100; % número de árboles
14
15
16 %% Leer parámetros de entrenamiento
17 fName = strcat(read_path, '/Descriptores_Prueba_Final.xlsx');
18 matr_descrip = readtable(fName, Sheet="Num");
19 ss = size(matr_descrip); %tamaño de la matriz, en este caso de 36x6
20 disp(strcat("Tabla de descriptores, shape=", num2str(ss(:,1)), ", ", num2str(ss(:,2))));
21
22 %% Entrenamiento y Validacion
23 % campos_a_evaluar = matr_descrip.Properties.VariableNames % todos los campos
24 [X_p, Y_p, X_t, Y_t] = CreaPruebaTest(matr_descrip, numero_elementos, elementos_test, campos_a_evaluar);
25
26 %% Configuración y obtención del RF
27 disp('Configuring Random Forest...');
28 myRF = TreeBagger(nArboles, X_p, Y_p, 'OOBPrediction', 'on');
```

Command Window

Tabla de descriptores, shape=50, 26
Numero de elementos por Tipo:5, Elementos para Test=2
Distribucion de Prueba=30, Test=20
Prueba: X=30, target=30
Prueba: X=20, target=20
Configuring Random Forest...

outputs =

1	1	2	2	3	3	4	4	5	5	6	6	7	7	8	8	9	10	10	10
---	---	---	---	---	---	---	---	---	---	---	---	---	---	---	---	---	----	----	----

ans =

1	1	2	2	3	3	4	4	5	5	6	6	7	7	8	8	9	9	10	10
---	---	---	---	---	---	---	---	---	---	---	---	---	---	---	---	---	---	----	----

Desempeño = 95%

>>

```
classif_ANNx.m x classif_RFx.m x +
64 X_v = table2array(X_v);
65
66 T_v = repmat([1 2 3 4 5 6 7 8 9 10],2,1); %Se crea una matriz con las etiquetas posibles para las muestras (2 filas, 9 columnas)
67 % target_v = T_v(:); %Se serializa T, Vector objetivo para la clasif supervisada
68 target_v = Test.ElementoID;
69 msg = strcat('Test: X= ', string(length(X_v)), ', target=', string(length(target_v)));
70 disp(msg);
71
72 %% Configuración y obtención del RF
73 disp('Configuring Random Forest...');
74 nArboles = 50; % número de árboles
75 myRF = TreeBagger(nArboles, XP, targetP, 'OOBPrediction','on');
76
77 %Respuesta del clasificador
78 res = myRF.predict(X_v);
79
80 %% resultados RF
81
82 resc = str2double(res);
83 outputs = resc' %outputs para presentar un vector fila con los resultados (como responde)
84 target_v' %... y compararlo con el objetivo (como debería responder)
85 %performance = perform(myRF, target', res)
86 % Evaluación del desempeño: Es mejor si se acerca a 100
87 eval = sum(resc==target_v)/length(target_v)*100;
88 disp(strcat('Desempeño = ', num2str(eval,2),'%'));
```

Command Window

```
Tabla de descriptores, shape=50, 26
Numero de elementos por Tipo:5, Elementos para Test=2
Distribucion de Prueba=30, Test=20
Prueba: X=30, target=30
Test: X=20, target=20
Configuring Random Forest...
```

outputs =

```
1 1 2 2 3 3 4 4 5 5 6 6 7 7 8 8 9 10 10 10
```

ans =

```
1 1 2 2 3 3 4 4 5 5 6 6 7 7 8 8 9 9 10 10
```

Desempeño = 95%

EXTRACCIÓN DE LA INFORMACIÓN Y ANÁLISIS DE DATOS

REDES NEURONALES –ENTRENADA

```
classif_ANNx.m x +
1 %%
2 % Script para CLASIFICACION CON REDES NEURONALES ARTIFICIALES
3 % Electiva: Procesamiento Digital de Imágenes
4 %
5 %% Limpiar el espacio de trabajo
6 clc; clear all; close all; % limpiar
7
8 %% Configuración inicial
9 read_path = '/MATLAB Drive/Published/Clasificacion';
10 numero_elementos = 5; %% numero de elementos por categoria
11 elementos_test = 2; %% numero de elementos del test <= numero_elementos
12 campos_a_evaluar = ["Area", "Circularity", "FilledArea"
13                    , "Perimeter", "MajorAxisLength"
14                    , "Puntas"];
15
16 hiddenLayerSize = [10 20 10];          %if I need more layers then I should write: [10,12,...,9]
17
18 %% Leer parámetros de entrenamiento
19 fName = strcat(read_path, '/Descriptores_Prueba_Final.xlsx');
20 matr_descrip = readtable(fName, Sheet="Num");
21 ss = size(matr_descrip); %tamaño de la matriz, en este caso de 36x6
22 disp(strcat("Tabla de descriptores, shape=", num2str(ss(:,1)), ", ", num2str(ss(:,2))))
23
24
25 %% Entrenamiento y Validacion
```

Command Window
Prueba: X=30, target=30
Prueba: X=20, target=20
Configuring Neural Network...

outputs =

4 1 2 2 3 3 4 4 5 5 6 8 7 8 8 8 9 10 10 10

ans =

1 1 2 2 3 3 4 4 5 5 6 6 7 7 8 8 9 9 10 10

eval =

80

Neural Network Training (22-Oct-2022 14:31:11)

Network Diagram

Training Results
Training finished: Met performance criterion ✓

Training Progress

Unit	Initial Value	Stopped Value	Target Value
Epoch	0	5	1000
Elapsed Time	-	00:00:02	-
Performance	26.7	0.000151	0.01
Gradient	163	0.253	1e-07
Mu	0.001	1e-06	1e+10
Validation Checks	0	0	6

Training Algorithms
Data Division: Random dividerand
Training: Levenberg-Marquardt trainlm
Performance: Mean Squared Error mse
Calculations: MEX

Training Plots

Performance

Training State

Error Histogram

Regression

Fit

ESTRACCIÓN DE LA INFORMACIÓN Y ANÁLISIS DE DATOS

REDES NEURONALES – ENTRENADA MEJORADA

```
classif_RFx.m x classif_ANNx.m x +
1 %%
2 % Script para CLASIFICACION CON REDES NEURONALES ARTIFICIALES
3 % Electiva: Procesamiento Digital de Imágenes
4 %
5 %% Limpiar el espacio de trabajo
6 clc; clear all; close all; % limpiar
7
8 %% Configuración inicial
9 read_path = '/MATLAB Drive/Published/Clasificacion';
10 numero_elementos = 5; %% numero de elementos por categoria
11 elementos_test = 2; %% numero de elementos del test <= numero_elementos
12 campos_a_evaluar = ["Area","Circularity", "FilledArea", "Perimeter", "MajorAxisLength", "Puntas"];
13
14 hiddenLayerSize = [10 20 10]; % If I need more layers then I should write: [10,12,...,9]
15
16 %% Leer parámetros de entrenamiento
17 fName = strcat(read_path,'/Descriptores_Prueba_Final.xlsx');
18 matr_descrip = readtable(fName, Sheet="Num");
19 ss = size(matr_descrip); %tamaño de la matriz, en este caso de 36x6
20 disp(strcat("Tabla de descriptores, shape=", num2str(ss(:,1)), ", ", num2str(ss(:,2))))
21
22
23 %% Entrenamiento y Validacion
24 % campos_a_evaluar = matr_descrip.Properties.VariableNames % todos los campos
25 [X_p, Y_p, X_t, Y_t] = CreaPruebaTest(matr_descrip, numero_elementos, elementos_test, campos_a_evaluar);
```

Command Window

```
80
outputs =
    1    1    1    1    3    3    4    4    5    5    6    6    7    7    8    8    9   10   10   10
ans =
    1 | 1    2    2    3    3    4    4    5    5    6    6    7    7    8    8    9    9   10   10
eval =
85
```

Neural Network Training (22-Oct-2022 14:50:27)

Network Diagram

Training Results
Training finished: Met performance criterion ✓

Training Progress

Unit	Initial Value	Stopped Value	Target Value
Epoch	0	10	1000
Elapsed Time	-	00:00:00	-
Performance	0.206	1.23e-23	1e-20
Gradient	3.08	3.24e-12	1e-07
Mu	0.001	1e-07	1e+10
Validation Checks	0	5	6

Training Algorithms
Data Division: Random dividerand
Training: Levenberg-Marquardt trainlm
Performance: Mean Squared Error mse
Calculations: MEX

Training Plots

Performance

Training State

Error Histogram

Regression

Fit

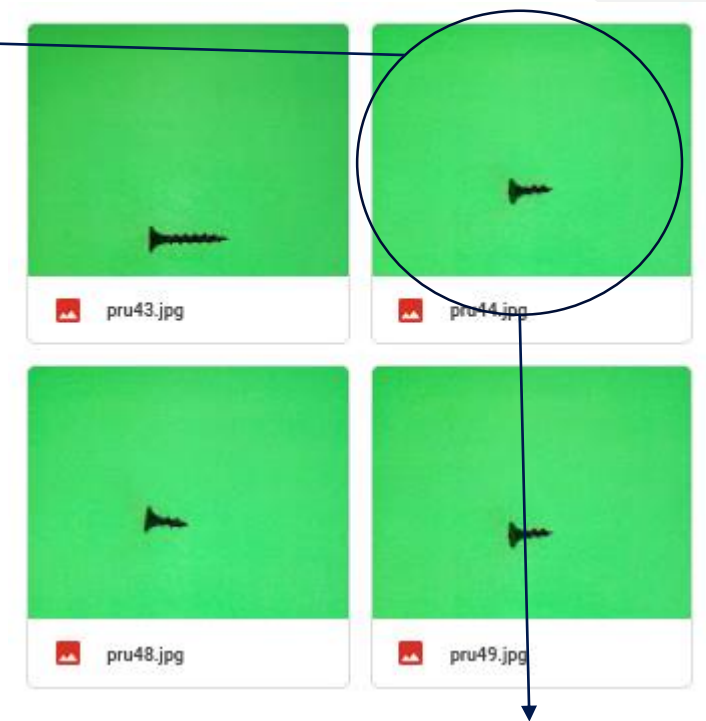
EXTRACCIÓN DE LA INFORMACIÓN Y ANÁLISIS DE DATOS

RANDOM FOREST

```
outputs =  
  1  1  2  2  3  3  4  4  5  5  6  6  7  7  8  8  9  10 10 10  
ans =  
  1  1  2  2  3  3  4  4  5  5  6  6  7  7  8  8  9  9 10 10  
Desempeño = 95%
```

REDES NEURONALES

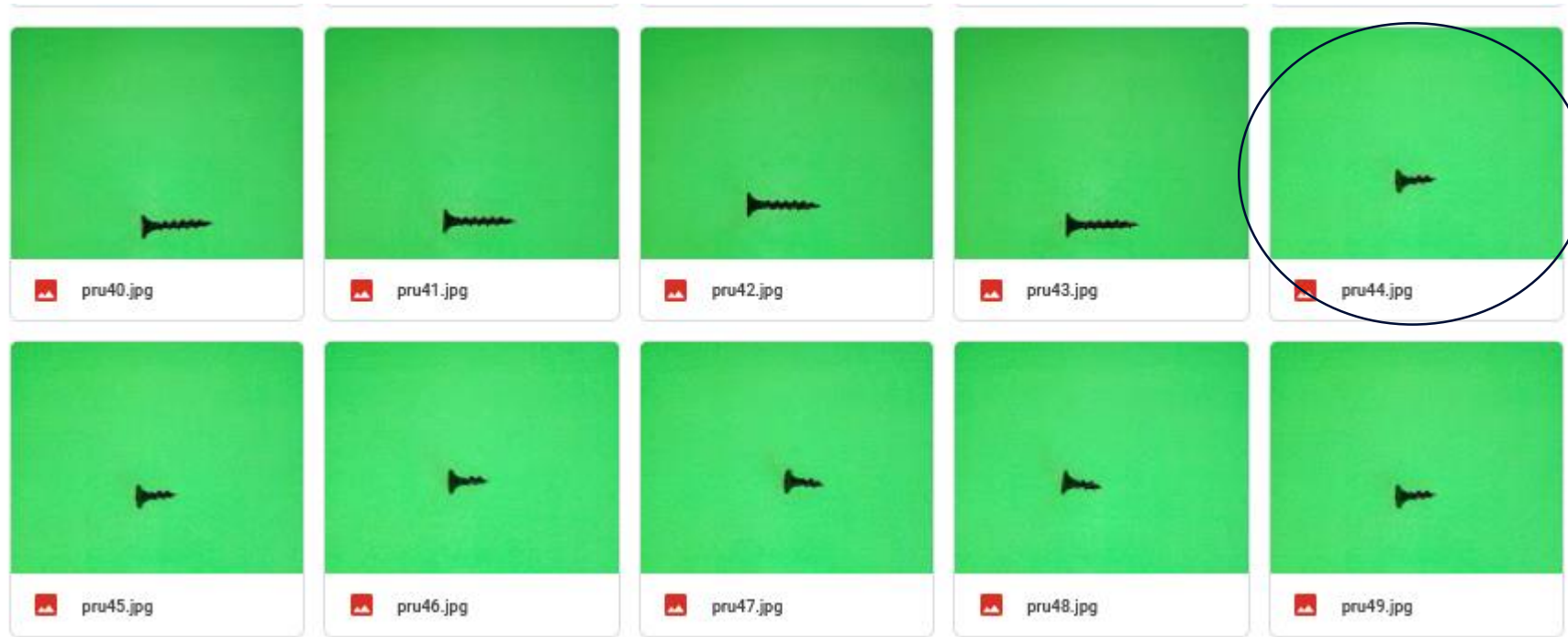
```
outputs =  
  1  1  1  1  3  3  4  4  5  5  6  6  7  7  8  8  9  10 10 10  
ans =  
  1  | 1  2  2  3  3  4  4  5  5  6  6  7  7  8  8  9  9 10 10  
eval =  
  85
```



Error por mala clasificación

ESTRACCIÓN DE LA INFORMACIÓN Y ANÁLISIS DE DATOS

REALMENTE



Cambiando la imagen
Manualmente

POSTERIOR AL REEMPLAZO DE LA IMAGEN

```
classif_RFx.m x +
1 %%
2 % Script para CLASIFICACION CON REDES RANDOM FOREST
3 % Electiva: Procesamiento Digital de Imágenes
4 %
5 %% Limpiar el espacio de trabajo
6 clc; clear all; close all; % limpiar
7
8 %% Configuración inicial
9 read_path = '/MATLAB Drive/Published/Clasificacion';
10 numero_elementos = 5; %% numero de elementos por categoria
11 elementos_test = 2; %% numero de elementos del test <= numero_elementos
12 campos_a_evaluar = ["Area", "Circularity", "FilledArea", "Perimeter", "MajorAxisLength", "Puntas"];
13
14 nArboles = 100; % número de árboles
15
16 %% Leer parámetros de entrenamiento
17 fName = strcat(read_path, '/Descriptores_entrenamiento.xlsx');
18 matr_descrip = readtable(fName);
19 ss = size(matr_descrip); %tamaño de la matriz, en este caso de 36x6
20 disp(strcat("Tabla de descriptores, shape=", num2str(ss(:,1)), ", ", num2str(ss(:,2))));
21
22 %% Entrenamiento y Validacion
23 % campos_a_evaluar = matr_descrip.Properties.VariableNames % todos los campos
24 [X_p, Y_p, X_t, Y_t] = CreaPruebaTest(matr_descrip, numero_elementos, elementos_test, campos_a_evaluar);
25
```

Command Window

Tabla de descriptores, shape=50, 26
Numero de elementos por Tipo:5, Elementos para Test=2
Distribucion de Prueba=30, Test=20
Prueba: X=30, target=30
Prueba: X=20, target=20
Configuring Random Forest...

outputs =

1	1	2	2	3	3	4	4	5	5	6	6	7	7	8	8	9	9	10	10
---	---	---	---	---	---	---	---	---	---	---	---	---	---	---	---	---	---	----	----

ans =

1	1	2	2	3	3	4	4	5	5	6	6	7	7	8	8	9	9	10	10
---	---	---	---	---	---	---	---	---	---	---	---	---	---	---	---	---	---	----	----

Desempeño = 100%

>>

EXTRACCIÓN DE LA INFORMACIÓN Y ANÁLISIS DE DATOS

REDES NEURONALES

POSTERIOR AL REEMPLAZO DE LA IMAGEN

classif_RFx.m x classif_ANNx.m x +

```
1 %%
2 % Script para CLASIFICACION CON REDES NEURONALES ARTIFICIALES
3 % Electiva: Procesamiento Digital de Imágenes
4 %
5 %% Limpiar el espacio de trabajo
6 clc; clear all; close all; % limpiar
7
8 %% Configuración inicial
9 read_path = '/MATLAB Drive/Published/Clasificacion';
10 numero_elementos = 5; %% numero de elementos por categoria
11 elementos_test = 2; %% numero de elementos del test <= numero_elementos
12 campos_a_evaluar = ["Area", "Circularity", "FilledArea", "Perimeter", "MajorAxisLength", "Puntas"];
13
14 hiddenLayerSize = [10 20 10]; %if I need more layers then I should write: [10,12,...,9]
15
16 %% Leer parámetros de entrenamiento
17 fName = strcat(read_path, '/Descriptores_entrenamiento.xlsx');
18 matr_descrip = readtable(fName);
19 ss = size(matr_descrip); %tamaño de la matriz, en este caso de 36x6
20 disp(strcat("Tabla de descriptores, shape=", num2str(ss(:,1)), ", ", num2str(ss(:,2))))
21
22
23 %% Entrenamiento y Validacion
24 % campos_a_evaluar = matr_descrip.Properties.VariableNames % todos los campos
25 [X_p, Y_p, X_t, Y_t] = CreaPruebaTest(matr_descrip, numero_elementos, elementos_test, campos_a_evaluar);
```

Command Window

ans =
1 1 2 2 3 3 4 4 5 5 6 6 7 7 8 8 9 9 10 10

Desempeño = 85%

outputs =
1 1 2 2 3 3 4 4 4 4 6 6 7 7 8 8 9 8 10 10

ans =
1 1 2 2 3 3 4 4 5 5 6 6 7 7 8 8 9 9 10 10

Desempeño = 85%
\\

Neural Network Training (22-Oct-2022 15:45:26)

Network Diagram

Training Results
Training finished: Reached minimum gradient ✓

Training Progress

Unit	Initial Value	Stopped Value	Target Value
Epoch	0	6	1000
Elapsed Time	-	00:00:00	-
Performance	0.0175	7.11e-26	1e-45
Gradient	1.31	5.1e-13	1e-07
Mu	0.001	1e-08	1e+10
Validation Checks	0	6	6

Training Algorithms
Data Division: Random dividerand
Training: Levenberg-Marquardt trainlm
Performance: Mean Squared Error mse
Calculations: MEX

Training Plots
Performance Training State
Error Histogram Regression
Fit

EXTRACCIÓN DE LA INFORMACIÓN Y ANÁLISIS DE DATOS

REDES NEURONALES

POSTERIOR AL REEMPLAZO DE LA IMAGEN

```
classif_RFx.m x classif_ANNx.m x +
1 %%
2 % Script para CLASIFICACION CON REDES NEURONALES ARTIFICIALES
3 % Electiva: Procesamiento Digital de Imágenes
4 %
5 %% Limpiar el espacio de trabajo
6 clc; clear all; close all; % limpiar
7
8 %% Configuración inicial
9 read_path = '/MATLAB Drive/Published/Clasificacion';
10 numero_elementos = 5; %% numero de elementos por categoria
11 elementos_test = 2; %% numero de elementos del test <= numero_elementos
12 campos_a_evaluar = ["Area", "Circularity", "FilledArea", "Perimeter", "MajorAxisLength", "Puntas"];
13
14 hiddenLayerSize = [10 20 10]; %%if I need more layers then I should write: [10,12,...,9]
15
16 %% Leer parámetros de entrenamiento
17 fName = strcat(read_path, '/Descriptores_entrenamiento.xlsx');
18 matr_descrip = readtable(fName);
19 ss = size(matr_descrip); %%tamaño de la matriz, en este caso de 36x6
20 disp(strcat("Tabla de descriptores, shape=", num2str(ss(:,1)), ", ", num2str(ss(:,2))));
21
22
23 %% Entrenamiento y Validacion
24 % campos_a_evaluar = matr_descrip.Properties.VariableNames % todos los campos
25 [X_p, Y_p, X_t, Y_t] = CreaPruebaTest(matr_descrip, numero_elementos, elementos_test, campos_a_evaluar);
```

Command Window

```
ans =
    1    1    2    2    3    3    4    4    5    5    6    6    7    7    8    8    9    9   10   10
Desempeño = 90%
outputs =
    1    1    2    2    3    3    4    4    5    6    6    6    7    7    8    8    8    9   10   10
ans =
    1    1    2    2    3    3    4    4    5    5    6    6    7    7    8    8    9    9   10   10
Desempeño = 90%
\`
```

Network Diagram

Training Results

Training finished: Met performance criterion ✓

Training Progress

Unit	Initial Value	Stopped Value	Target Value
Epoch	0	4	1000
Elapsed Time	-	00:00:00	-
Performance	0.0285	2.25e-20	1e-10
Gradient	1.97	1.31e-09	1e-07
Mu	0.001	1e-07	1e+10
Validation Checks	0	4	100

Training Algorithms

Data Division: Random dividerand

Training: Levenberg-Marquardt trainlm

Performance: Mean Squared Error mse

Calculations: MEX

Training Plots

Performance Training State

Error Histogram Regression

Fit

ENLACE DE LOS ARCHIVOS DE TODO EL PROYECTO

<https://drive.google.com/drive/folders/1jKnRrr1gnOj-MvHq8m-RdVE8S5qc6zQ7?usp=sharing>