

Implementación de códigos en software

PUNTO 1

- a. Desarrolle el siguiente código en el script de Matlab y guárdelo como “aprendizaje”. (Asegúrese de estar trabajando en la carpeta que ya se ha creado). Para esta unidad el estudiante debe trabajar la base de datos de entrenamiento y la base de datos de prueba (**las dos bases de datos serán proporcionadas por el tutor, en la apertura del foro**)

El código a utilizar es el que ya se implementó en **la etapa 2**, en el **PUNTO 1**. La combinación de numpixel y umbral, debe ser **única** para todas las imágenes y debe ser la que segmente mejor los ID terminados en 1 y los ID terminados en 4. Es decir, el estudiante no debe cambiar el valor del umbral y del numpixel en cada imagen, sino debe ser un mismo valor de estas dos variables para todas las imágenes proporcionadas por el tutor.

Nota1: Si no está trabajando en la carpeta donde se encuentra la imagen guardada, al realizar el código le saldrá error.

Nota2: Para que el código sea válido, debe tener una línea con su nombre, año, periodo académico y semestre actual que está cursando el estudiante (según plan de estudios).

```
% Nombre: Paola Mateus
% Año: XXXX
% Periodo: XXXX
% Semestre XX

[Lo num]=bwlabel(Filtro1);
prop=regionprops(Lo);
figure;
imshow(Lo)

ultimo_numero = imcrop(Filtro1,prop(end).BoundingBox);
figure;
imshow(ultimo_numero)
prop_ultimo_numero=regionprops(ultimo_numero,'Centroid','Circularity')
```

- Explique cada línea de código.
- ¿Cómo funciona y que datos obtienen de la función “bwlabel”?
- ¿Qué valor se obtiene en las variables “Lo” y en “num”?

- d. ¿Cómo funciona y que datos obtienen de la función “regionprops”?

Nota 3: El filtro utilizado para este experimento es el generado después de la binarización (**Etapa 2, PUNTO 1**), si el estudiante utiliza el filtro que se obtuvo después de la dilatación no obtendrá el puntaje correcto de este experimento.

PUNTO 2

- a. Cada estudiante debe obtener las siguientes características de cada una de las imágenes de la base de datos de **entrenamiento**. Las características las debe guardar en una matriz de Excel donde se tenga la siguiente información (ver Tabla 1):

Tabla 1

Datos a recopilar en el proceso de entrenamiento para cada una de las imágenes.

	Centroide en X	Centroide en Y	Circularidad	Clasificación SVM	Clasificación Experto
Imagen de entrenamiento ID 1				1	
Imagen de entrenamiento ID 2				1	
Imagen de entrenamiento ID 3				1	
Imagen de entrenamiento ID 4				1	
Imagen de entrenamiento ID 5				0	
Imagen de entrenamiento ID 6				0	
Imagen de entrenamiento ID 7				0	

El estudiante debe anexar el resultado de cada imagen procesada con un pantallazo de toda la ventana de Matlab, donde se observe: ComandWindows, Workspace, y Current Folder.

De acuerdo a la investigación que usted ha realizado sobre máquinas de soporte vectorial, sabe que la clasificación está dada en binario, por lo tanto, 1 será nuestra clasificación para la imagen de los ID terminados en 1, y 0 para imagen de los ID terminados en 4.

- Guarde la tabla anterior como “entrenamiento.xlsx”

PUNTO 3

- Desarrolle el siguiente código en el script de Matlab creado en el punto 1. (Asegúrese de estar trabajando en la carpeta que ya se ha creado).

Nota1: Si no está trabajando en la carpeta donde se encuentra la imagen guardada, al realizar el código le saldrá error.

Nota2: Para que el código sea válido, debe tener una línea con su nombre.

Cada estudiante debe obtener las siguientes características de cada una de las imágenes de la base de datos de **prueba**. Las características las debe guardar en una matriz de Excel que se llame “prueba.xlsx” donde se tenga la siguiente información (ver Tabla 2):

Tabla 2

Información obtenida para las imágenes de prueba, luego de correr la SMV

	Centroide en X	Centroide en Y	Circularidad	Clasificación SVM	Clasificación Experto
Imagen de prueba - ID 1					
Imagen de prueba - ID2					
Imagen de prueba - ID 3					
Imagen de prueba - ID 4					

Imagen de prueba - ID 5					
Imagen de prueba - ID 6					
Imagen de prueba - ID 7					

```
%%
% Nombre: Paola Mateus
% Año: XXXX
% Periodo: XXXX
% Semestre XX
clc
clear all
close all
datos=xlsread('datos.xlsx');
caracteristicas=datos(1:end,1:3);
entrenamiento=datos(1:end,4);

%Entrenamiento Máquina - Paola Mateus
svmStruct =
fitcsvm(caracteristicas,entrenamiento,'Standardize',true,'KernelFunction',
'linear','KernelScale','auto');

prueba=xlsread('prueba.xlsx');
xprueba=prueba(1:end,1:3);

clasificacion=predict(svmStruct,xprueba)
```

- El estudiante debe incluir en el documento de entrega un pantallazo de la tabla de las características con las que entreno la máquina, es decir las contenidas en entrenamiento.xlsx y un pantallazo de la tabla que contiene las características nuevas contenidas en prueba.xlsx.
- Explique cada línea del código.

PUNTO 4

- Clasificador de umbral o perceptrón: con dos características (centroide en Y, y circularidad), el estudiante realizará el clasificador por umbral. El estudiante debe investigar cómo realizar este clasificador, anexar el código y resultados obtenidos. Diligenciar la

Tabla 3 para tener claridad de los valores a utilizar en el diseño del clasificador por umbral.

Tabla 3

Datos previos para diseñar el clasificador por umbral

	Centroide en Y	Circularidad
Imagen de prueba – ID 1		
Imagen de prueba – ID 2		
Imagen de prueba – ID 3		
Imagen de prueba – ID 4		
Imagen de prueba – ID 5		
Imagen de prueba – ID 6		
Imagen de prueba – ID 7		

Nota: Este clasificador no se realiza con el mismo código de la SVM, ustedes deben crear el código usando condicionales “if”.