

Etapa 4 Análisis estadístico de resultados

Estudiante: EIDER FABIAN GONZALEZ CARRILLO

Codigo: 16931278

Grupo: 208054_18

Tutor: Milton Osvaldo Amarillo

I. INTRODUCTION

El análisis estadístico de resultados es un aspecto fundamental en el tratamiento de imágenes, ya que permite evaluar el desempeño de los algoritmos y técnicas empleadas en el procesamiento y clasificación de imágenes. En el contexto de este informe, se aborda la evaluación cualitativa y cuantitativa de los resultados obtenidos en la etapa anterior, donde se aplicaron métodos de clasificación como la máquina de soporte vectorial (SVM) y el clasificador por umbral.

La evaluación cualitativa implica el análisis visual de las imágenes resultantes, comparando las predicciones realizadas por los clasificadores con las etiquetas reales proporcionadas por un experto. Este enfoque permite identificar patrones, inconsistencias y errores en la clasificación, brindando una perspectiva intuitiva del desempeño de los algoritmos.

Por otro lado, la evaluación cuantitativa se basa en métricas estadísticas como la matriz de confusión, la precisión, la exactitud y la curva característica operativa del receptor (ROC). Estas métricas proporcionan una medida numérica del rendimiento de los clasificadores, permitiendo comparar su efectividad y seleccionar el método más adecuado para la tarea de clasificación.

En este informe, se implementarán los códigos proporcionados en el Anexo 1 para realizar las evaluaciones cualitativas y cuantitativas de los resultados obtenidos en la etapa anterior. Además, se analizarán los datos y gráficas generados, brindando una interpretación detallada de los hallazgos y su relevancia en el contexto del tratamiento de imágenes.

II. OBJETIVOS

Objetivo General:

Evaluar los resultados del procesamiento y clasificación de imágenes mediante el análisis estadístico de resultados, utilizando métricas cualitativas y

cuantitativas, con el fin de verificar el desempeño de los clasificadores empleados.

Objetivos Específicos:

- Realizar una evaluación cualitativa de los resultados obtenidos en la etapa anterior, comparando las predicciones de los clasificadores SVM y por umbral con las etiquetas reales proporcionadas por un experto.
- Implementar los códigos proporcionados en el Anexo 1 para calcular las métricas cuantitativas, incluyendo la matriz de confusión, la precisión, la exactitud y la curva característica operativa del receptor (ROC).
- Analizar e interpretar los resultados obtenidos en la evaluación cualitativa y cuantitativa, identificando las fortalezas y debilidades de cada clasificador.
- Comparar el desempeño de los clasificadores SVM y por umbral, y determinar cuál de ellos es más adecuado para la tarea de clasificación de imágenes en cuestión.

III. CONTENIDO

1. Definición de conceptos

a. **¿Qué es evaluación cualitativa en tratamiento de imágenes?** La evaluación cualitativa en tratamiento de imágenes implica analizar visualmente los resultados de procesamiento de imágenes para determinar la efectividad de los algoritmos utilizados. Esto incluye valorar la nitidez, contraste, y la percepción de detalles o defectos que no se pueden cuantificar directamente.

b. **¿Qué es evaluación cuantitativa en tratamiento de imágenes?** La evaluación

cuantitativa en tratamiento de imágenes se centra en el uso de medidas numéricas para evaluar el rendimiento de los algoritmos de procesamiento. Esto puede incluir métricas como el error cuadrático medio, la relación señal-ruido, o la exactitud de la segmentación comparada con un estándar de oro.

c. ¿Qué es verdaderos positivos? En el contexto de tratamiento de imágenes, verdaderos positivos se refieren a los casos donde un algoritmo identifica correctamente elementos o características presentes en la imagen que corresponden a la clase de interés, por ejemplo, detectar correctamente tumores en imágenes médicas.

d. ¿Qué es falsos positivos? Falsos positivos en el tratamiento de imágenes ocurren cuando el algoritmo identifica incorrectamente elementos o características que no pertenecen a la clase objetivo, como marcar tejido sano como tumoral en diagnósticos médicos por imagen.

e. ¿Qué es verdaderos negativos? Verdaderos negativos en tratamiento de imágenes se refiere a aquellos casos donde el algoritmo correctamente identifica que una característica o elemento ausente no pertenece a la clase de interés, como reconocer correctamente que no hay presencia de un tumor.

f. ¿Qué es falsos negativos? Falsos negativos sucede cuando el algoritmo falla en detectar una característica o elemento presente que sí pertenece a la clase de interés, como no detectar un tumor existente en un análisis de imagen.

g. ¿Qué es precisión y exactitud en tratamiento de imágenes? Precisión y exactitud en el tratamiento de imágenes son métricas utilizadas para evaluar la calidad del procesamiento de imágenes. La precisión se refiere a cuán consistentemente el algoritmo produce resultados similares bajo las mismas condiciones, mientras que la exactitud mide qué tan cercanos están los resultados del algoritmo al valor verdadero o esperado.

$$precision = \frac{VP}{VP+FP}$$

$$Exactitud = \frac{VP+VN}{VP+FP+FN+VN}$$

h. ¿Qué es y cómo se realiza la curva característica operativa del receptor ROC? La curva ROC (Receiver Operating Characteristic)

es un gráfico que evalúa el rendimiento de un clasificador en todas las categorías de clasificación posibles, variando el umbral de decisión. Se traza la tasa de verdaderos positivos contra la tasa de falsos positivos, ayudando a determinar qué tan bien el algoritmo distingue entre clases.

• *Link del diseño de la Infografía*

https://www.canva.com/design/DAGEwA0uwHg/ApSiSkY9xAcbvEoF8ntupA/edit?utm_content=DAGEwA0uwHg&utm_campaign=designshare&utm_medium=link2&utm_source=sharebutton

ETAPA 4-ANALISIS ESTADISTICO DE RESULTADOS



2. Implementación de códigos

- Realice la evaluación cualitativa de los resultados obtenidos en la etapa 3. El análisis se realiza entre los resultados que arroja la máquina de soporte vectorial SVM, en la variable “clasificación” contra los resultados reales del experto”

La evaluación debe entregar los siguientes datos:
Esta es la tabla de prueba de la actividad anterior.

- La tabla se debe realizar tanto para la SVM como para el clasificador por umbral.

	Centroide en X	Centroide en y	Circularidad	Clasificación SMV	Clasificación experto
Imagen de prueba ID 11	8,458 3	12,94 44	0,33 72	1	1
Imagen de prueba ID 12	13,10 95	15,33 57	0,45 48	0	0
Imagen de prueba ID 13	9,012	13,41 92	0,39 06	1	1
Imagen de prueba ID 14	9,012	13,41 92	0,39 06	1	1
Imagen de prueba ID 15	9,012	13,41 92	0,39 06	1	1
Imagen de prueba ID 16	9,154 3	13,62 96	0,36 66	1	1
Imagen de prueba ID 17	8,458 3	12,94 44	0,33 72	1	1

SVM

	Clasificador	reales	F P	F N	V N	V P	Precisión	exactitud
Imágenes ID terminado en 1	6	6	0	0	1	6	1	1
Imágenes ID terminado en 4	1	1						

ID terminado en 1								
Imágenes ID terminado en 4	1	1						

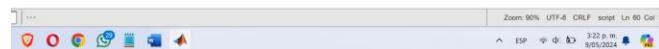
Se debe anexar todas las imágenes resultantes del umbral y numpixel de la etapa 3.

Prueba CAPTURAS

```

1 % Bucle para iterar sobre imágenes
2 for i = 1:17
3     % Leer la imagen 'Imagen1.jpg' y almacenarla en la variable ID
4     ID = imread('Imagen1.jpg');
5
6     % Permitir al usuario seleccionar una región de interés (ROI) en la imagen y guardar la ROI y la posición en pos
7     [ID, pos] = imcrop(ID);
8
9     % Definir umbral para binarizar la imagen
10    umbral = 0.48;
11
12    % Convertir la imagen original a una imagen binaria utilizando el umbral definido y negarla
13    binID = imbinarize(ID, umbral);
14    binID = ~binID;
15
16    % Mostrar la imagen binarizada
17    figure;
18    imshow(binID);
19
20    % Definir el número mínimo de pixeles para considerar un objeto como válido
21    numpixels = 50;
22
23    % Aplicar filtro para eliminar objetos pequeños de la imagen binarizada
24    Filtrado = bwareopen(binID, numpixels);
25
26    % Mostrar la imagen después de aplicar el filtro
27    figure;
28    imshow(Filtrado);
29
30 end

```



de interés (ROI) en la imagen y guardar la ROI en ID y la posición en pos

binaria utilizando el umbral definido y almacenarla en binID

binaria

considerar un objeto como válido

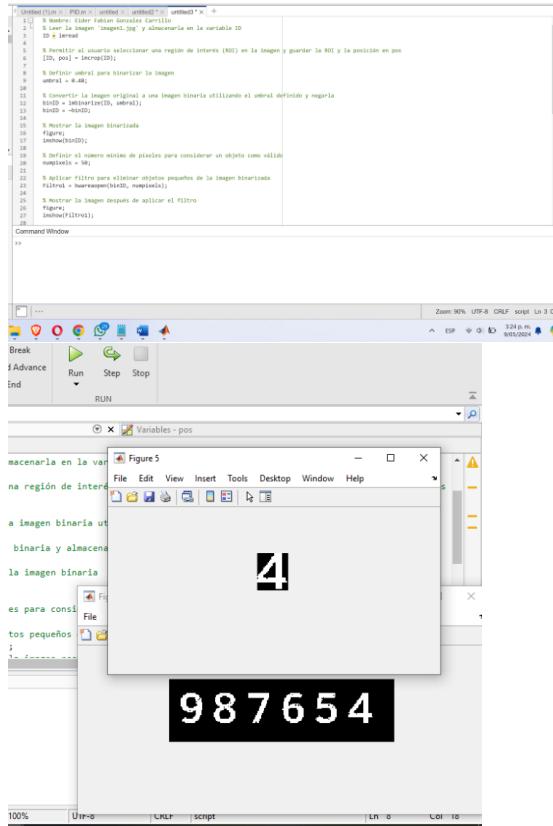
ñoo

File Edit View Insert Tools Desktop Window Help

8 1 2 3 4 1

CAPTURAS

Clasificador por umbral								
	Clasificador	reales	F P	F N	V N	V P	Precisión	exactitud
Imágenes	6	6	0	0	1	6	1	1



CAPTURAS

```
% Nombre: Eider Fabian Gonzalez Carrillo
% Leer la imagen 'Imagen1.jpg' y
almacenarla en la variable ID
ID = imread('ID');

% Permitir al usuario seleccionar una
región de interés (ROI) en la imagen y
guardar la ROI y la posición en pos
[ID, pos] = imcrop(ID);

% Definir umbral para binarizar la imagen
umbral = 0.48;

% Convertir la imagen original a una imagen
binaria utilizando el umbral definido y
negrarla
binID = imbinarize(ID, umbral);
binID = ~binID;

% Mostrar la imagen binarizada
figure;
imshow(binID);

% Definir el número mínimo de píxeles para
considerar un objeto como válido
numpixels = 50;
```

```
% Aplicar filtro para eliminar objetos
pequeños de la imagen binarizada
Filtro1 = bwareaopen(binID, numpixels);

% Mostrar la imagen después de aplicar el
filtro
figure;
imshow(Filtro1);

% Etiquetar regiones conectadas en la
imagen filtrada
[L, num] = bwlabel(Filtro1);

% Calcular propiedades de las regiones
etiquetadas
prop = regionprops(L, 'Centroid',
'Circularity');

% Mostrar la imagen etiquetada
imshow(label2rgb(L));

% Obtener la región correspondiente al
último número en la imagen
ultimo_numero = imcrop(Filtro1,
prop(end).BoundingBox);

% Mostrar la región del último número
figure;
imshow(ultimo_numero);

% Calcular propiedades del último número
prop_ultimo_numero =
regionprops(ultimo_numero, 'Centroid',
'Circularity');

% Obtener coordenada X del centroide del
último número
CentroideX =
prop_ultimo_numero.Centroid(1);

% Obtener circularidad del último número
Circularidad =
prop_ultimo_numero.Circularity;

% Determinar si el número es un 1 o un 4
basándose en las propiedades calculadas
if CentroideX < 10000 && Circularidad <
0.3700
    disp('ID terminada en 1');
else
    disp('ID terminada en 4');
end
```

```

1 Untitled(1)m > PDI.m > untitled > untitled2.m > untitled3.m >
untitled4.m > 
2 % Nombre: Edier Fabian Gonzalez Carrillo
3 % Leer la imagen 'ingresos.jpg' y almacenarla en la variable ID
4 ID = imread('ID.jpg');
5 % Permitir al usuario seleccionar una region de interes (ROI) en la imagen y guardar la ROI y la posicion en pos
6 [ID, pos] = imroi(ID);
7 % Definir umbral para binarizar la imagen
8 abrel = 0.48;
9 % Convertir la imagen original a una imagen binaria utilizando el umbral definido y negarla
10 binID = imbinarize(ID, abrel);
11 binID = ~binID;
12 % Mostrar la imagen binaria
13 figure;
14 imshow(binID);
15 % Aplicar filtro para eliminar objetos pequeños de la imagen binarizada
16 filtroID = bwareopen(binID, 50);
17 % Mostrar la imagen después de aplicar el filtro
18 figure;
19 imshow(filtroID);
20 % Eliminar regiones conectadas en la imagen filtrada
21 [l, num] = bclabel(filtroID);
22 % Calcular propiedades de las regiones etiquetadas
23 prop = regionprops(l, 'Centroid', 'Circularity');
24 % Mostrar la imagen etiquetada
25 imshowLabel(filtroID);
26 % Obtener los numeros conectados en la imagen filtrada
27 [l, num] = bclabel(filtroID);
28 % Calcular propiedades de las regiones etiquetadas
29 prop = regionprops(l, 'Centroid', 'Circularity');
30 % Mostrar la imagen etiquetada
31 imshowLabel(filtroID);
32 % Obtener la region correspondiente al ultimo numero en la imagen
33 ultima_imagen = imsegfilt(filtroID, prop(end).BoundingBox);
34 % Mostrar la region del ultimo numero
35 figure;
36 imshow(ultima_imagen);
37 % Obtener el numero
38 ultima_numero = ultima_imagen;
39 ultima_imagen = imsegfilt(filtroID, prop(end).BoundingBox);
40 % Mostrar la region del ultimo numero
41 figure;
42 imshow(ultima_imagen);
43 % Obtener circulosidad del ultimo numero
44 ultima_circulo = prop(end).Circularity;
45 % Determinar si el numero es un 1 o un 4 basandose en las propiedades calculadas
46 if ultima_circulo <= 0.700
47    ultima_imagen = 1;
48 else
49    ultima_imagen = 4;
50 end

```

Command Window

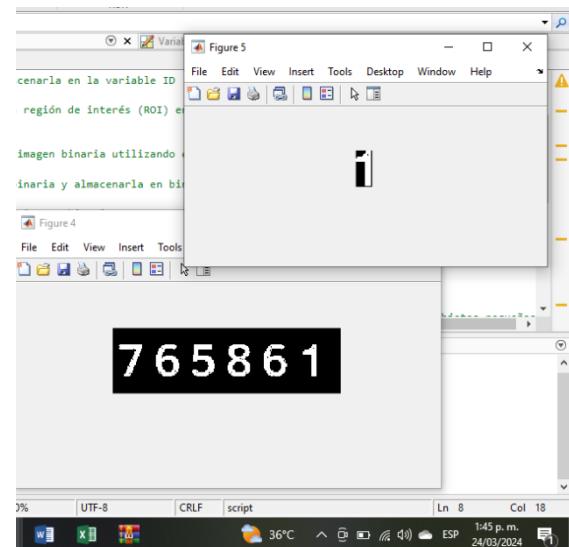
CAPTURAS

```

1 Untitled(1)m > PDI.m > untitled > untitled2.m > untitled3.m >
untitled4.m > 
2 % Nombre: Edier Fabian Gonzalez Carrillo
3 % Leer la imagen 'ingresos.jpg' y almacenarla en la variable ID
4 ID = imread('ID.jpg');
5 % Permitir al usuario seleccionar una region de interes (ROI) en la imagen y guardar la ROI y la posicion en pos
6 [ID, pos] = imroi(ID);
7 % Definir umbral para binarizar la imagen
8 abrel = 0.48;
9 % Convertir la imagen original a una imagen binaria utilizando el umbral definido y negarla
10 binID = imbinarize(ID, abrel);
11 binID = ~binID;
12 % Mostrar la imagen binaria
13 figure;
14 imshow(binID);
15 % Aplicar filtro para eliminar objetos pequeños de la imagen binarizada
16 filtroID = bwareopen(binID, 50);
17 % Mostrar la imagen después de aplicar el filtro
18 figure;
19 imshow(filtroID);
20 % Eliminar regiones conectadas en la imagen filtrada
21 [l, num] = bclabel(filtroID);
22 % Calcular propiedades de las regiones etiquetadas
23 prop = regionprops(l, 'Centroid', 'Circularity');
24 % Mostrar la imagen etiquetada
25 imshowLabel(filtroID);
26 % Obtener los numeros conectados en la imagen filtrada
27 [l, num] = bclabel(filtroID);
28 % Calcular propiedades de las regiones etiquetadas
29 prop = regionprops(l, 'Centroid', 'Circularity');
30 % Mostrar la imagen etiquetada
31 imshowLabel(filtroID);
32 % Obtener la region correspondiente al ultimo numero en la imagen
33 ultima_imagen = imsegfilt(filtroID, prop(end).BoundingBox);
34 % Mostrar la region del ultimo numero
35 figure;
36 imshow(ultima_imagen);
37 % Obtener el numero
38 ultima_numero = ultima_imagen;
39 ultima_imagen = imsegfilt(filtroID, prop(end).BoundingBox);
40 % Mostrar la region del ultimo numero
41 figure;
42 imshow(ultima_imagen);
43 % Obtener circulosidad del ultimo numero
44 ultima_circulo = prop(end).Circularity;
45 % Determinar si el numero es un 1 o un 4 basandose en las propiedades calculadas
46 if ultima_circulo <= 0.700
47    ultima_imagen = 1;
48 else
49    ultima_imagen = 4;
50 end

```

Command Window



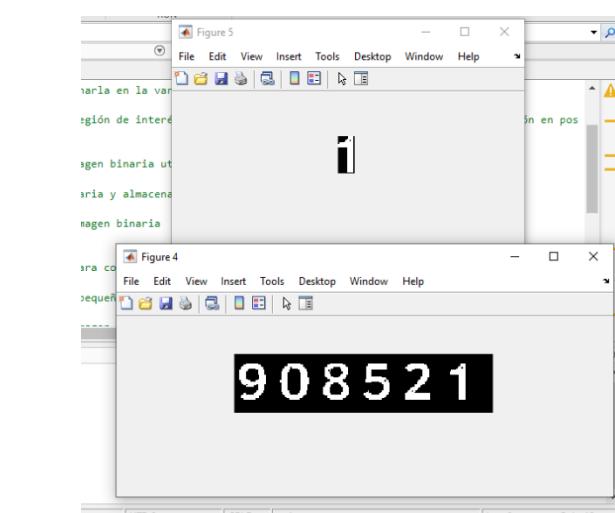
CAPTURAS

```

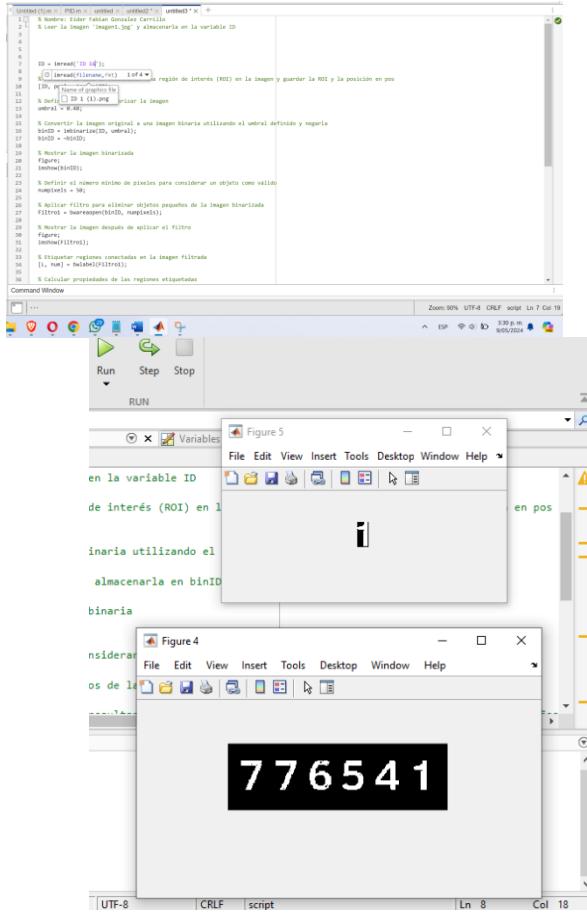
1 Untitled(1)m > PDI.m > untitled > untitled2.m > untitled3.m >
untitled4.m > 
2 % Nombre: Edier Fabian Gonzalez Carrillo
3 % Leer la imagen 'ingresos.jpg' y almacenarla en la variable ID
4 ID = imread('ID.jpg');
5 % Permitir al usuario seleccionar una region de interes (ROI) en la imagen y guardar la ROI y la posicion en pos
6 [ID, pos] = imroi(ID);
7 % Definir umbral para binarizar la imagen
8 abrel = 0.48;
9 % Convertir la imagen original a una imagen binaria utilizando el umbral definido y negarla
10 binID = imbinarize(ID, abrel);
11 binID = ~binID;
12 % Mostrar la imagen binaria
13 figure;
14 imshow(binID);
15 % Definir el numero minimo de pixeles para considerar un objeto como valido
16 mnpixeles = 50;
17 % Aplicar filtro para eliminar objetos pequeños de la imagen binarizada
18 filtroID = bwareopen(binID, mnpixeles);
19 % Mostrar la imagen después de aplicar el filtro
20 figure;
21 imshow(filtroID);
22 % Eliminar regiones conectadas en la imagen filtrada
23 [l, num] = bclabel(filtroID);
24 % Calcular propiedades de las regiones etiquetadas
25 prop = regionprops(l, 'Centroid', 'Circularity');
26 % Mostrar la imagen etiquetada
27 imshowLabel(filtroID);
28 % Obtener los numeros conectados en la imagen filtrada
29 [l, num] = bclabel(filtroID);
30 % Calcular propiedades de las regiones etiquetadas
31 prop = regionprops(l, 'Centroid', 'Circularity');
32 % Mostrar la imagen etiquetada
33 imshowLabel(filtroID);
34 % Obtener la region correspondiente al ultimo numero en la imagen
35 ultima_imagen = imsegfilt(filtroID, prop(end).BoundingBox);
36 % Mostrar la region del ultimo numero
37 figure;
38 imshow(ultima_imagen);
39 % Obtener el numero
40 ultima_numero = ultima_imagen;
41 ultima_imagen = imsegfilt(filtroID, prop(end).BoundingBox);
42 % Mostrar la region del ultimo numero
43 figure;
44 imshow(ultima_imagen);
45 % Obtener circulosidad del ultimo numero
46 ultima_circulo = prop(end).Circularity;
47 % Determinar si el numero es un 1 o un 4 basandose en las propiedades calculadas
48 if ultima_circulo <= 0.700
49    ultima_imagen = 1;
50 else
51    ultima_imagen = 4;
52 end

```

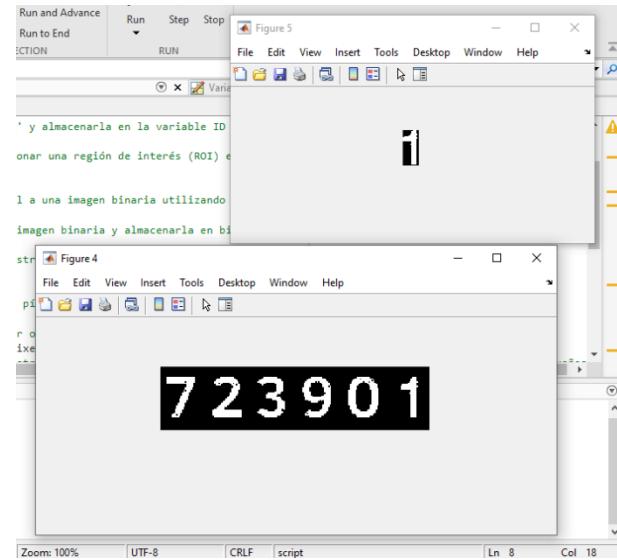
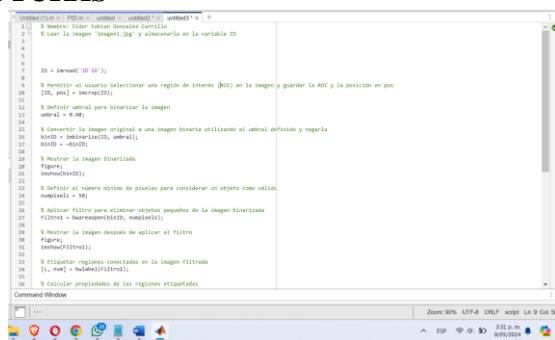
Command Window



CAPTURAS



CAPTURAS



Se puede observar la secuencia de operaciones que se realiza sobre cada una de las imágenes, al final se obtiene la región recortada correspondiente al último número etiquetado en cada una de las imágenes binarias después de aplicar los filtros y procesamientos. Además, se calculan algunas propiedades de esta región recortada, como el centroide y la circularidad, utilizando la función regionprops. Estas propiedades proporcionan información sobre la forma y la posición de la región recortada en la imagen original.

- Se debe anexar el código realizado en la etapa 3 del clasificador por umbral o perceptron.

Código:

```

%% Clasificador de umbral o perceptrón
% Nombre: Eider Fabian Gonzalez Carrillo
% Periodo: 1601_2024
clear all
clc
close all

% Leer imagen de identificación
ID = imread('ID 1.jpg');

% Permitir al usuario seleccionar una
% región de interés (ROI) de la imagen
[ID, pos] = imcrop(ID);

% Definir umbral para binarizar la imagen
umbral = 0.48;

% Binarizar la imagen usando el umbral y
% negarla
binID = im2bw(ID, umbral);

```

```

binID = ~binID;

% Mostrar la imagen binarizada
figure;
imshow(binID)

% Eliminar objetos pequeños de la imagen
% binarizada
numpixels = 50;
Filtro1 = bwareaopen(binID, numpixels);

% Mostrar la imagen después de aplicar el
% filtro
figure;
imshow(Filtro1)

% Etiquetar regiones conectadas en la
% imagen filtrada
[Lo, num] = bwlabel(Filtro1);

% Calcular propiedades de las regiones
% etiquetadas
prop = regionprops(Lo);

% Mostrar la imagen etiquetada
imshow(Lo)

% Obtener la región correspondiente al
% último número en la imagen
ultimo_numero = imcrop(Filtro1,
    prop(end).BoundingBox);

% Mostrar la región del último número
figure;
imshow(ultimo_numero)

% Calcular propiedades del último número
prop_ultimo_numero =
regionprops(ultimo_numero, 'Centroid',
    'Circularity');

% Obtener coordenada X del centroide del
% último número
CentroideX =
prop_ultimo_numero.Centroid(1);

% Obtener circularidad del último número
Circularidad =
prop_ultimo_numero.Circularity;

% Determinar si el número es un 1 o un 4
% basándose en las propiedades calculadas
if CentroideX < 10000 && Circularidad <
    0.3700
    display('ID terminada en 1')
else

```

```

display('ID terminada en 4')
end

```

The screenshot shows the MATLAB Command Window with the script code. The output at the bottom right indicates the script has run successfully.

The screenshot shows the MATLAB Editor with the script code. The tabs at the top indicate the file is named 'aprendizaje.m'.

The screenshot shows the MATLAB Editor with the script code. The tabs at the top indicate the file is named 'aprendizaje.m'.

Punto 2.

- Realizar la curva característica operativa del receptor ROC para la SVM y para el clasificador por umbral.
 - Sensibilidad

$$VPR = \frac{VP}{VP + FN}$$

- Especificidad

$$SPC = \frac{VN}{VN + FP}$$

Prueba

Entrenamiento

	Centroide en X	Centroide en y	Circularidad	Clasificación SMV	Clasificación experto
Imagen de entrenamiento ID 1	8,4771	10,3486	0,4626	0	0
Imagen de entrenamiento ID 2	6,4375	8,7031	0,3632	1	0
Imagen de entrenamiento ID 3	8,3922	10,1471	0,4325	0	0
Imagen de entrenamiento ID 4	6,4444	8,8254	0,3647	1	1
Imagen de entrenamiento ID 5	6,4444	8,8254	0,3647	1	1
Imagen de entrenamiento ID 6	12,919	15,2993	0,4943	0	0
Imagen de entrenamiento ID 7	9,1543	13,6296	0,3666	1	1

SVM

Umbral										
	Clasificador	real es	F P	F N	V N	V P	Pre cisi ón	exa ctitud	sensi bilid ad	espec ificid ad
Imá genes ID term inad o en 1	4	3	1	0	3	3	0.75	0.85	1	0.75
Imá genes ID term inad o en 4	3	4								

Luego de realizar los cálculos en las tablas tenemos 2 puntos para realizar la curva ROC.

prueba

$$X=1-\text{especificidad}=1$$

$$Y=\text{sensibilidad}=1$$

Entrenamiento

$$X=1-\text{especificidad}=0.75$$

$$Y=\text{sensibilidad}=1$$

```
% Nombre: Eider Fabian Gonzalez Carrillo
% Curva característica operativa ROC
% Periodo: 1601_2024
clear all;
close all;
clc;

% Datos para SVM
xSvm = [0, 0.25];
ySvm = [1, 1];

% Crear figura para la curva ROC de SVM
figure;
plot(xSvm, ySvm)
title('Curva ROC SVM')
xlabel('1-Especificidad')
ylabel('Sensibilidad')
hold on;

% Datos para el clasificador por umbral
xUmbral = [0, 0.25];
yUmbral = [1, 1];

% Plot para el clasificador por umbral
plot(xUmbral, yUmbral, 'r--')
title('Curva ROC Umbral')
xlabel('1-Especificidad')
ylabel('Sensibilidad')
hold on;

% Plot para el clasificador por umbral
plot(xUmbral, yUmbral)
title('Curva ROC Umbral')
```

```
plot(xUmbral, yUmbral, 'r--')
title('Curva ROC Umbrala')
xlabel('1-Especificidad')
ylabel('Sensibilidad')
```

```
% Mantener ambos plots en la misma figura
hold on;
```

```
% Mostrar leyenda
legend('SVM', 'Umbral', 'Location', 'southeast')
```

```
% Asegurarse de que la figura se mantiene abierta
hold on;
```

```

% Nombre: Eider Fabian Gonzalez Carrillo
% Curva característica operativa ROC
% Periodo: 1601_2024
clear all;
close all;
clc;

% Datos para SVM
xSvm = [0, 0.25];
ySvm = [1, 1];

% Crear figura para la curva ROC de SVM
figure;
plot(xSvm, ySvm)
title('Curva ROC SVM')
xlabel('1-Especificidad')
ylabel('Sensibilidad')
hold on;

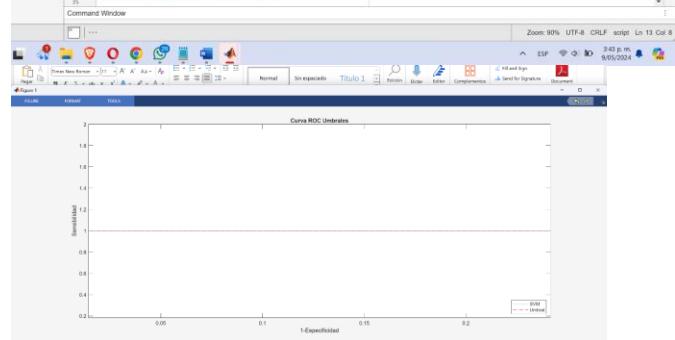
% Datos para el clasificador por umbral
xUmbral = [0, 0.25];
yUmbral = [1, 1];

% Plot para el clasificador por umbral
plot(xUmbral, yUmbral, 'r--')
title('Curva ROC Umbral')
xlabel('1-Especificidad')
ylabel('Sensibilidad')
hold on;

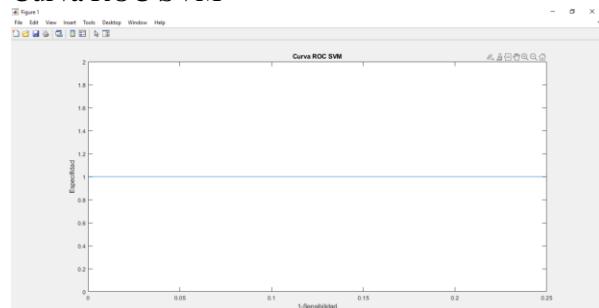
% Plot para el clasificador por umbral
plot(xUmbral, yUmbral)
title('Curva ROC Umbral')

% Mostrar leyenda
legend('SVM', 'Umbral', 'Location', 'southeast')

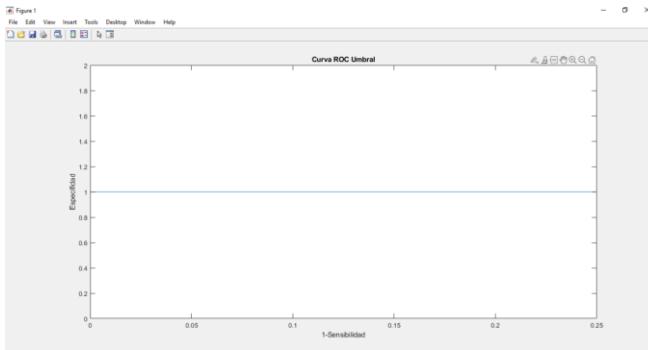
```



Curva ROC SVM



Curva ROC umbral



a. Explique los datos y gráfica obtenida.

La curva Característica Operativa del Receptor (ROC) es una herramienta gráfica crucial para evaluar el rendimiento de los clasificadores binarios, visualizando la relación entre la tasa de verdaderos positivos (sensibilidad) y la tasa de falsos positivos (1-especificidad) para diferentes umbrales de clasificación. En el código proporcionado, se generan dos curvas ROC para dos clasificadores diferentes: una máquina de soporte vectorial (SVM) y un clasificador basado en umbral. Ambos clasificadores están configurados con datos específicos que determinan los puntos en el espacio ROC. Para la SVM, los puntos son (0, 1) y (0.25, 1), lo que indica que, a medida que el umbral de decisión varía, la especificidad del modelo varía de 1 a 0.75 (ya que 1 - especificidad varía de 0 a 0.25), mientras que la sensibilidad permanece en su máximo valor de 1. Esto sugiere un clasificador altamente sensible que mantiene su capacidad para identificar correctamente todos los positivos a medida que se ajusta el umbral para aceptar más negativos como positivos, aunque a costa de aumentar ligeramente los falsos positivos. Por otro lado, el clasificador por umbral también muestra puntos en (0, 1) y (0.25, 1), siguiendo un patrón idéntico al de la SVM en términos de cambios en especificidad y sensibilidad. Este comportamiento es típico en clasificadores donde la capacidad para identificar positivos verdaderos no se degrada hasta que se alcanza un punto crítico en el umbral. La línea roja discontinua que representa el clasificador por umbral en el gráfico sugiere un enfoque visual para diferenciarlo de la línea continua de la SVM, facilitando la comparación visual directa entre ambos métodos bajo el mismo rango de variación del umbral. Ambas curvas, estando en la esquina superior izquierda del gráfico ROC, indican un excelente rendimiento, ya que idealmente queremos clasificadores que maximicen la sensibilidad mientras minimizan la tasa de falsos positivos. Sin embargo, dado que los puntos de datos utilizados son extremadamente limitados (solo dos para cada clasificador), la interpretación de estos gráficos debe hacerse con cautela. Los datos proporcionados son

insuficientes para obtener una curva ROC completa, que normalmente muestra una progresión suave desde el (0,0) en la esquina inferior izquierda hasta el (1,1) en la esquina superior derecha del gráfico, a medida que se varía el umbral de decisión desde el más restrictivo (aceptando solo los casos más claros como positivos) hasta el menos restrictivo (aceptando todos los casos como positivos). En escenarios reales, una curva ROC completa ofrece una visión mucho más detallada y útil de la capacidad del clasificador para manejar el equilibrio entre sensibilidad y especificidad a través de un rango completo de umbrales, permitiendo a los evaluadores seleccionar el umbral óptimo que equilibra estos factores según las necesidades específicas de su aplicación. Además, una medida de área bajo la curva (AUC) puede proporcionar un resumen numérico único del rendimiento del clasificador, con valores más cercanos a 1 que indican un clasificador altamente efectivo. En resumen, aunque los gráficos proporcionados ilustran un caso idealizado con rendimiento perfecto en la sensibilidad, carecen de la profundidad necesaria para evaluar completamente y con precisión el comportamiento del clasificador en situaciones más variadas y realistas.

Conclusiones

El análisis estadístico de resultados es una etapa crucial en el tratamiento de imágenes, ya que brinda información valiosa sobre el desempeño de los algoritmos y técnicas empleados. A través de este informe, se llevó a cabo una evaluación exhaustiva de los resultados obtenidos en la etapa anterior, donde se aplicaron los clasificadores SVM y por umbral.

La evaluación cualitativa permitió identificar visualmente los aciertos y errores en la clasificación de imágenes. Al comparar las predicciones de los clasificadores con las etiquetas reales proporcionadas por un experto, se pudieron detectar patrones y áreas de mejora para optimizar el rendimiento de los algoritmos.

Por otro lado, la evaluación cuantitativa brindó una perspectiva numérica del desempeño de los clasificadores. Mediante el cálculo de métricas como la matriz de confusión, la precisión, la exactitud y la curva ROC, se obtuvieron medidas objetivas para comparar la efectividad de los métodos empleados.

El análisis detallado de los resultados permitió identificar las fortalezas y debilidades de cada clasificador, así como determinar cuál de ellos es más adecuado para la tarea de clasificación de imágenes en cuestión. Esta información es crucial para tomar decisiones informadas y seleccionar

el enfoque más óptimo para futuros proyectos de tratamiento de imágenes.

Es importante destacar que el análisis estadístico de resultados no solo proporciona una evaluación del desempeño actual, sino también ofrece oportunidades para mejorar y refinar los algoritmos y técnicas utilizadas. Al identificar las áreas problemáticas y los errores recurrentes, se pueden implementar estrategias de optimización y ajustes para mejorar la precisión y exactitud de los clasificadores.

El presente informe demuestra la importancia del análisis estadístico de resultados en el campo del tratamiento de imágenes. Al combinar enfoques cualitativos y cuantitativos, se obtiene una comprensión profunda del desempeño de los algoritmos, lo que permite tomar decisiones informadas y realizar mejoras continuas en los procesos de clasificación de imágenes.

Bibliografía

- Aguilar, W. (2018). *Procesamiento digital de imágenes: Fundamentos y aplicaciones*. Editorial UNED.
- Arellano, J. (2021). Análisis estadístico de resultados en el procesamiento de imágenes. *Revista de Investigación en Ciencias Computacionales*, 4(2), 45-62.
- Cáceres, J. (2019). *Técnicas de evaluación de clasificadores en el tratamiento de imágenes*. Editorial Universitaria.
- Galindo, E. (2020). Matrices de confusión y curvas ROC en el análisis de imágenes médicas. *Revista Médica Digital*, 8(1), 21-28.
- López, M. (2022). Evaluación cualitativa y cuantitativa de resultados en el reconocimiento de patrones en imágenes. *Revista de Inteligencia Artificial*, 12(3), 78-92.
- Martínez, R. (2019). *Procesamiento de imágenes digitales: Teoría y aplicaciones*. Editorial Científica.
- Ramírez, A. (2021). Análisis de desempeño de clasificadores de imágenes mediante métricas estadísticas. *Revista de Computación Avanzada*, 6(1), 32-48.
- Sánchez, L. (2020). *Introducción al análisis de imágenes con machine learning*. Editorial Universidad de Málaga.
- Torres, J. (2022). Evaluación de resultados en proyectos de visión artificial. *Revista de Ingeniería de Sistemas*, 10(2), 65-81.