

OBJETIVO:

Realizar un detector de piel con lógica difusa apoyándose en el modelo HSV (Hue, Saturation, Value) en Matlab.

CONSIDERACIONES PREVIAS:

Modelo HSV

El modelo HSV representa los colores en tres componentes (figuras 1 y 2):

H (Matiz): Está en un rango de 0 a 360 grados (figura 2).

S (Saturación): Se encuentra en un rango de 0 a 100, donde 0 es sin saturación (poco color) y 100 es alta saturación (mucho color).

V (Brillo o iluminación): Se encuentra en un rango de 0 a 100, donde 0 es sin iluminación y 100 es alta iluminación.

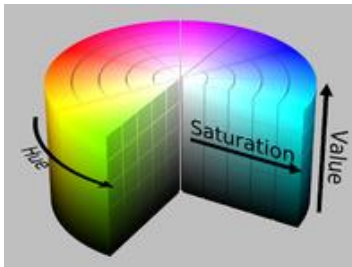


Figura 2. Se muestra la gráfica de cómo cambian los valores H, S y V para formar algún color. (https://en.wikipedia.org/wiki/HSL_and_HSV)

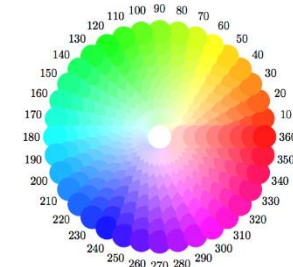


Figura 1. Se muestra la gráfica de H, el cual define el matiz de la imagen. (<https://i.stack.imgur.com/BfTXy.png>)

HSV en Matlab

`rgb2hsv()`. Esta función convierte un mapa de colores RGB a uno en HSV. Devuelve 3 matrices normalizadas, una para H, una para S y otra para V.

Lógica difusa

Se requiere un sistema de inferencia difuso con tres entradas y una salida para la detección de la piel. La entrada es una imagen a color en formato HSV y la salida es una imagen en escala de grises (sólo tres componentes: negro “No-piel”, un tono de gris “puede ser piel” y blanco “Sí-piel”).

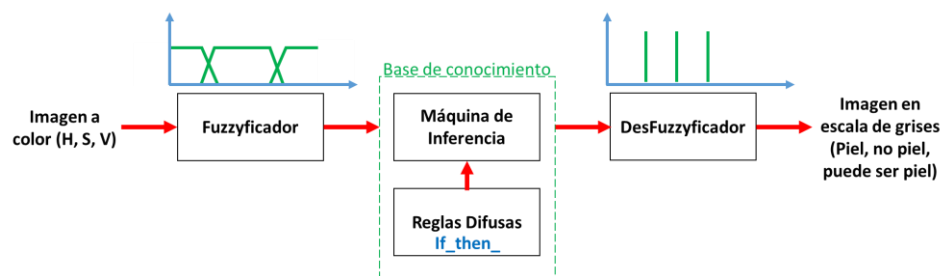


Figura 3. Sistema de inferencia difuso

DESARROLLO Y RESULTADOS:

Para poder distinguir entre piel y no piel se busca los intervalos de valores de H, S y V dentro de los cuales se encuentra la del ser humano. Debido a que existe un amplio rango de tonalidades de piel, lo cual hace compleja la detección de la misma, se opta por seleccionar un tono promedio en la población mexicana.

Con Matlab se analizan cinco imágenes de rostros con tonalidad similar y se obtienen histogramas para conocer los rangos de H, S y V. Posteriormente, basándose en los histogramas, se definen los conjuntos difusos para H, S y V. A continuación se muestran los resultados.



Figura 4. Imágenes de rostros utilizadas para obtener la base de conocimiento (conjuntos H, S, V)

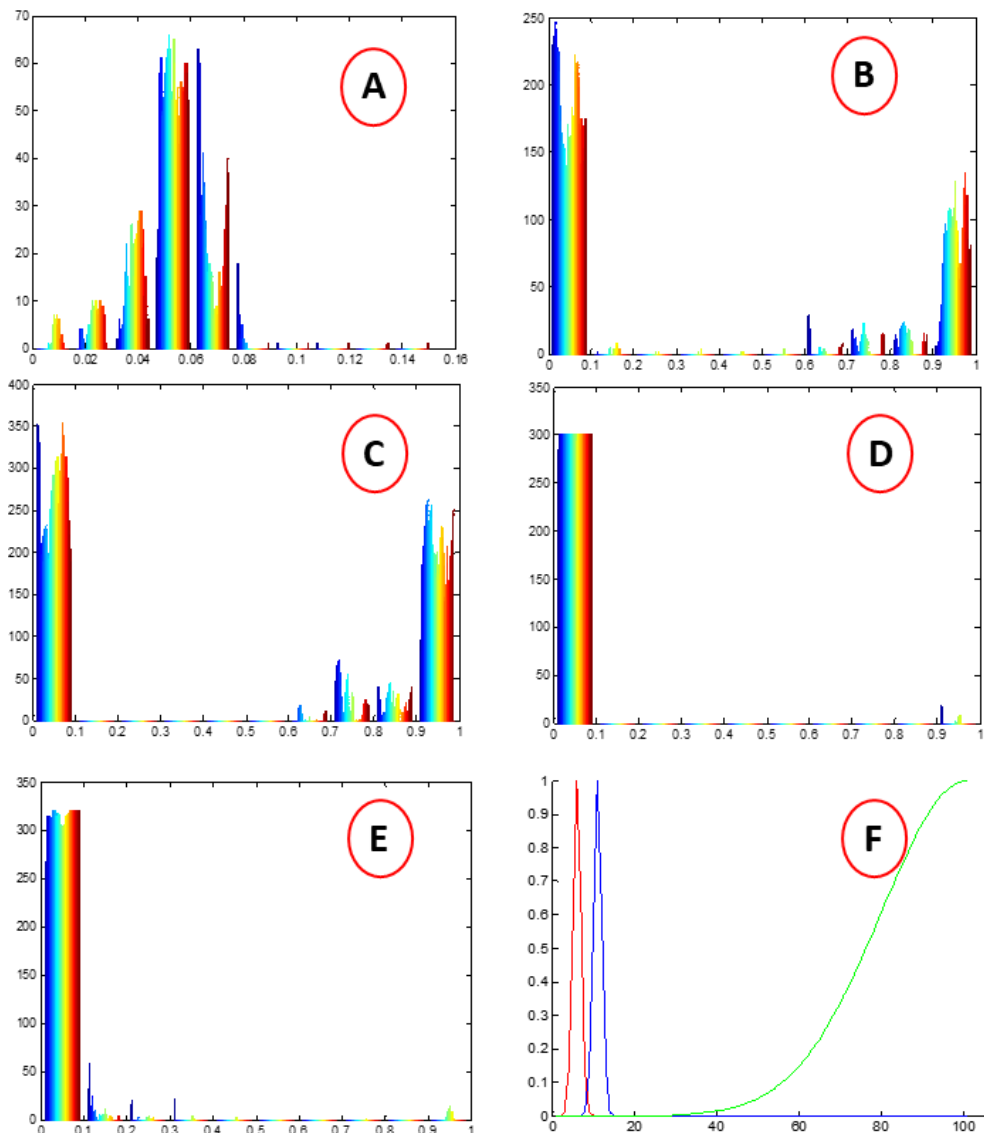


Figura 5. Matiz (H). A, B, C, D y E son los histogramas de las imágenes. F muestra los conjuntos difusos creados en base a los histogramas

La figura 5 muestra los histogramas de las imágenes de la figura 4. Se observa que la piel se encuentra en el intervalo de $0.02 \leq H \leq 0.1$, es decir, si se hace la conversión de 0 a 360° se tiene que la piel está entre el rojo claro y el naranja ($7.2^\circ \leq H \leq 36^\circ$) figura 2.

En la figura 5F se muestra los conjuntos difusos para H, el conjunto rojo (H1) representa “naranja” (en este conjunto cae el color de la piel), el conjunto azul (H2) representa “naranja claro” (donde es poco probable que sea piel si se detecta ese color) y el conjunto verde (H3) representa “otro color”, es decir que este conjunto corresponde al resto de colores que no pueden ser piel.

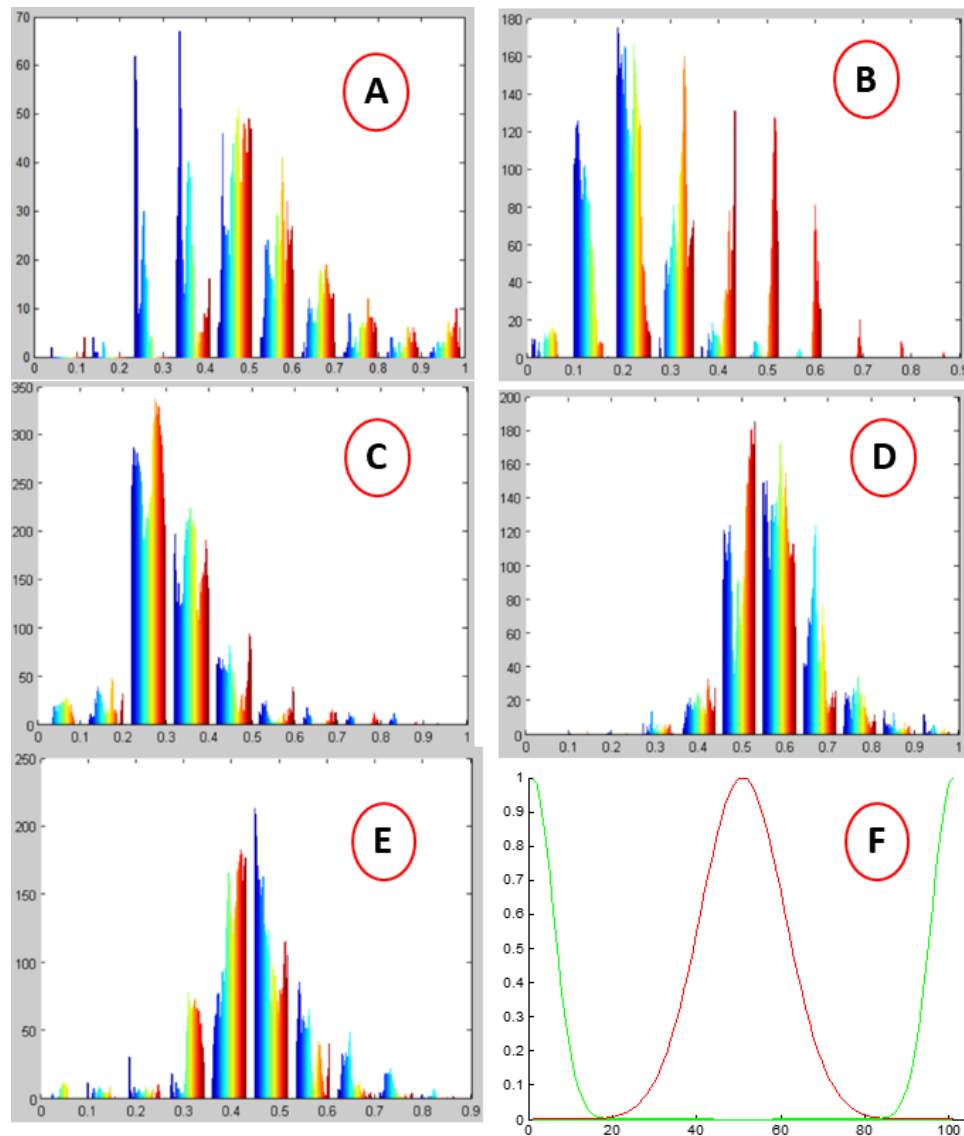


Figura 6. Saturación (S). A, B, C, D y E son los histogramas de las imágenes. F muestra los conjuntos difusos creados en base a los histogramas

En la figura 6 A, B, C, D y E se observa que la saturación de la piel está entre $0.2 \leq S \leq 0.8$, con un máximo alrededor de 0.5. La figura 6F muestra los conjuntos difusos para la saturación, Los conjuntos de los extremos S1 y S3 (verdes) representan colores “sin saturación” y con “alta

saturación”, es decir, si se detecta ese nivel de saturación se puede asegurar que no se está en presencia de piel. El conjunto rojo (S2) está en el intervalo de saturación promedio para la piel.

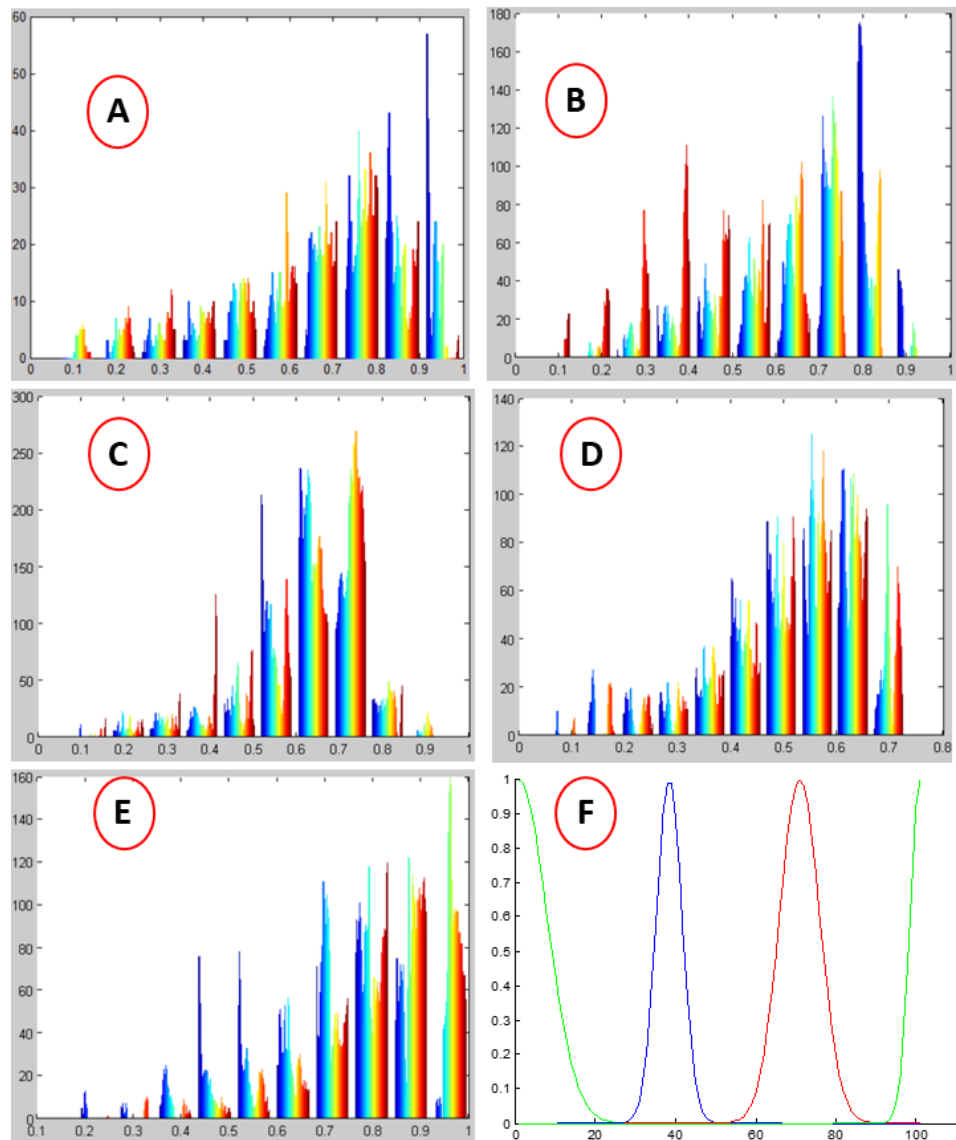


Figura 7. Iluminación (V). A, B, C, D y E son los histogramas de las imágenes. F muestra los conjuntos difusos creados en base a los histogramas

La figura 7F muestra los conjuntos difusos para la iluminación o brillo. Los conjuntos de los extremos V1 y V3 (verdes) representan valores de “no luz” y “sólo luz” por lo que si se detecta una imagen en estos intervalos se descarta que sea piel. Se realizan dos conjuntos que caen dentro de la iluminación “ideal” para determinar que se está detectando piel: El conjunto azul “visible 1” (V2) está en un intervalo de poca probabilidad de ser piel, el conjunto rojo “visible 2” (V3).

A continuación se muestra una tabla que contiene todas las reglas para el detector de piel.

Regla							Single tone
1	if	H1	S1	V1	then	No piel	P1
2	if	H1	S1	V2	then	No piel	P1
3	if	H1	S1	V3	then	Puede ser	P2
4	if	H1	S1	V4	then	No piel	P1
5	if	H1	S2	V1	then	No piel	P1
6	if	H1	S2	V2	then	Puede ser	P2
7	if	H1	S2	V3	then	Sí piel	P3
8	if	H1	S2	V4	then	No piel	P1
9	if	H1	S3	V1	then	No piel	P1
10	if	H1	S3	V2	then	No piel	P1
11	if	H1	S3	V3	then	No piel	P1
12	if	H1	S3	V4	then	No piel	P1
13	if	H2	S1	V1	then	No piel	P1
14	if	H2	S1	V2	then	No piel	P1
15	if	H2	S1	V3	then	No piel	P1
16	if	H2	S1	V4	then	No piel	P1
17	if	H2	S2	V1	then	No piel	P1
18	if	H2	S2	V2	then	No piel	P1
19	if	H2	S2	V3	then	Puede ser	P2
20	if	H2	S2	V4	then	No piel	P1
21	if	H2	S3	V1	then	No piel	P1
22	if	H2	S3	V2	then	No piel	P1
23	if	H2	S3	V3	then	No piel	P1
24	if	H2	S3	V4	then	No piel	P1
25	if	H3	S1	V1	then	No piel	P1
26	if	H3	S1	V2	then	No piel	P1
27	if	H3	S1	V3	then	No piel	P1
28	if	H3	S1	V4	then	No piel	P1
29	if	H3	S2	V1	then	No piel	P1
30	if	H3	S2	V2	then	No piel	P1
31	if	H3	S2	V3	then	No piel	P1
32	if	H3	S2	V4	then	No piel	P1
33	if	H3	S3	V1	then	No piel	P1
34	if	H3	S3	V2	then	No piel	P1
35	if	H3	S3	V3	then	No piel	P1
36	if	H3	S3	V4	then	No piel	P1

Debido a que el sistema de inferencia tiene tres entradas, para calcular la salida se obtiene el mínimo de las membresías de estos, se multiplica cada mínimo por su correspondiente single tone, se suma los resultados y se divide todo entre la suma de los mínimos, es decir:

$$salida = \frac{\sum_j (\mu_{j \min} * P)}{\sum_j (\mu_{j \min})}$$

Para simplificar los cálculos (y el código para Matlab) se selecciona el single tone P1=0, es decir que si no se detecta piel se tendrá un pixel negro a la salida, quedando la ecuación anterior como

$$salida = \frac{(\mu_{3 \min} * P2) + (\mu_{6 \min} * P2) + (\mu_{7 \min} * P3) + (\mu_{19 \min} * P2)}{\mu_{3 \min} + \mu_{6 \min} + \mu_{7 \min} + \mu_{19 \min}}$$

Donde

$$\mu_{3 \min} = \min(\mu_{H3}, \mu_{S3}, \mu_{V3})$$

$$\mu_{6 \min} = \min(\mu_{H6}, \mu_{S6}, \mu_{V6})$$

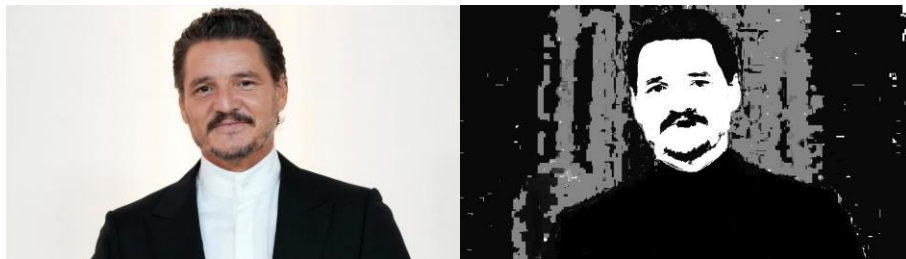
$$\mu_{7 \min} = \min(\mu_{H7}, \mu_{S7}, \mu_{V7})$$

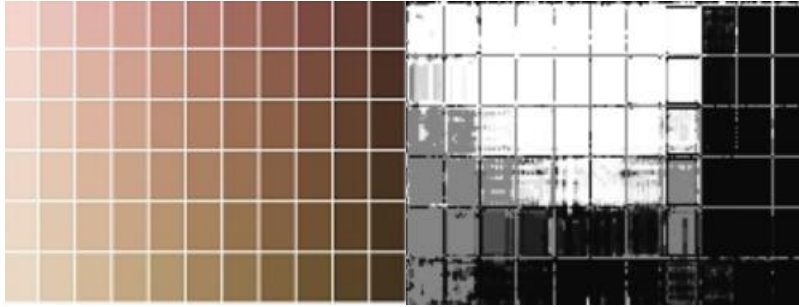
$$\mu_{19 \min} = \min(\mu_{H19}, \mu_{S19}, \mu_{V19})$$

$$P2 = 10 \quad y \quad P3 = 255$$

Esto se realiza para cada uno de los pixeles de la imagen.

Para comprobar el correcto funcionamiento del algoritmo, se utilizan diversas imágenes. Los resultados se muestran a continuación.





CÓDIGO EN MATLAB:

```

clc, close all, clear all;

Ima = imread('C:\Users\USER\Documents\ Detector de piel\Ima\Im1.jpg');

Imahsv=rgb2hsv(Ima);
%+++++++FIGURA 1+++++++
figure, imshow(Ima)

[fil,col3]=size(Imahsv);
col=col3/3;%Porque son 3 capas

h=Imahsv(:, :,1);
s=Imahsv(:, :,2);
v=Imahsv(:, :,3);

H1=0.05; dH1=0.01; H2=0.1; dH2=0.01; H3=1; dH3=0.21;
S1=0; dS1=0.05; S2=0.5; dS2=0.10; S3=1; dS3=0.05;
V1=0; dV1=0.07; V2=0.375; dV2=0.032; V3=0.7; dV3=0.05; V4=1; dV4=0.025;

p1=0; p2=10; p3=255;

sal = zeros(fil,col);
for i=1:fil
    for j=1:col
        %pix = double(Ima(i,j))+1;
        H_in=h(i,j);
        S_in=s(i,j);
        V_in=v(i,j);

        [Hue1 Hue2 Hue3] = Membresia(H_in,H1, dH1, H2, dH2, H3, dH3);

        [Sat1 Sat2 Sat3] = Membresia(S_in,S1, dS1, S2, dS2, S3, dS3);

        [Val1 Val2 Val3 Val4] = Membresia4(V_in,V1, dV1, V2, dV2, V3, dV3, V4,dV4);

        R3=min(Hue1,min(Sat1,Val1));
        R6=min(Hue1,min(Sat2,Val2));
        R7=min(Hue1,min(Sat2,Val3));
        R19=min(Hue2,min(Sat2,Val3));

        if (R3==0 && R6==0 && R7==0 && R19==0)
            UNO=1;
        else
            UNO=0;
        end

        sal(i,j)=( (R3*p2)+(R6*p2)+(R7*p3)+(R19*p2)) / (R3+R6+R7+R19+UNO);

    end
end

```

```

end

%*****FIGURA 2*****
figure, imshow(uint8(sal))

%Gráfica del HUE
k=0;
for H_in =0:0.01:1
    [Hache1 Hache2 Hache3] = Membresia(H_in,H1, dH1, H2, dH2, H3, dH3);
    k = k+1;
    uh1(k)=Hache1;
    uh2(k)=Hache2;
    uh3(k)=Hache3;
end
figure, hold on, plot(uh1,'r'),plot(uh2,'b'),plot(uh3,'g'),hold off

%Gráfica del Saturacion
l=0;
for S_in =0:0.01:1
    [Satur1 Satur2 Satur3] = Membresia(S_in,S1, dS1, S2, dS2, S3, dS3);
    l = l+1;
    us1(l)=Satur1;
    us2(l)=Satur2;
    us3(l)=Satur3;
end
figure, hold on, plot(us1,'g'),plot(us2,'r'),plot(us3,'g'),hold off

%Gráfica del Valor
l=0;
for V_in =0:0.01:1
    [Valor1 Valor2 Valor3 Valor4] = Membresia4(V_in,V1, dV1, V2, dV2, V3, dV3, V4,dV4);
    l = l+1;
    uv1(l)=Valor1;
    uv2(l)=Valor2;
    uv3(l)=Valor3;
    uv4(l)=Valor4;
end
figure, hold on, plot(uv1,'g'),plot(uv2,'b'),plot(uv3,'r'),plot(uv4,'g'),hold off

-----

function[con1 con2 con3]=Membresia(dato,a1, d1, a2, d2, a3, d3)%'dato' de entrada

M=dato;

con1=exp((- (M-a1)^2) / (2* (d1^2)));
con2=exp((- (M-a2)^2) / (2* (d2^2)));
con3=exp((- (M-a3)^2) / (2* (d3^2)));

-----

function[con1 con2 con3 con4]=Membresia4(dato,a1, d1, a2, d2, a3, d3, a4, d4)%'dato' de
entrada

M=dato;

con1=exp((- (M-a1)^2) / (2* (d1^2)));
con2=exp((- (M-a2)^2) / (2* (d2^2)));
con3=exp((- (M-a3)^2) / (2* (d3^2)));
con4=exp((- (M-a4)^2) / (2* (d4^2)));

```


CONCLUSIONES:

Se puede observar que el detector de piel trabaja de manera satisfactoria, ya que no sólo tiene un buen desempeño con las imágenes de “entrenamiento”.

Debido a que la piel sólo se encuentra en unos pocos rangos de H, S y V, muchas de las reglas nunca aportan nada al cálculo de la imagen de salida, dejando una operación simple pero muy eficiente.

REFERENCIAS

<https://www.mathworks.com/help/matlab/ref/rgb2hsv.html>