ANEXO 2
(código)

entrenamiento.m

```
%entrenamiento

entrenamiento1
close all
entrenamiento2
close all
entrenamiento3
close all
entrenamiento4
close all
```

entrenamiento1.m

```
%entrenamiento1

global matriz1
global objeto1

%el programa lee la imagen
I1=imread('foto90.jpg');
%se muestra la imagen original en una figura
%imshow(I1)
%title('Imagen Original')
%convertir la imagen a intensidad (grayscale)
I2=I1(:,:,1);
%procedimiento para obtener el contorno
%I3=wiener2(I2);
E=edge(I2,'canny');
E2=bwperim(E);

%procesamiento para cerrar el contorno
B2=bwmorph(E2, 'close');
B=bwfill(E2, 'holes');
B1=bwmorph(B, 'remove');
B3=bwmorph(B1, 'thicken');
%se muestran los contornos de las imagenes
figure,imshow(B3)
title('Contornos de la Imagen')

%seleccion
BS=bwselect(B3, 8);
figure, imshow(BS)
title('Contorno seleccionado')

%matriz de distancias
M=zeros(size(BS)); %se crea una matriz del mismo tamaño que la matriz original
[r,c]=size(BS); %se obtiene el numero de renglones y columnas de la matriz original
%se hace un barrido que cambia los valores
for j=1: r,
   for i=1: c,
      pixel=BS(j,i);
```

```
      if (pixel==0)
        M(j,i)=255;
      else M(j,i)=0;
      end
   end
end

%transformada de distancia
[ren,col]=size(M);
V1=zeros(1,6);
V2=zeros(1,6);
for j=2: ren-1,
   for i=2: col-1,
      pixel1=(1.4142 + M((j-1),(i+1)));
      V1(1)=pixel1;
      pixel2=(1 + M((j-1),i));
      V1(2)=pixel2;
      pixel3=(1.4142 + M((j-1),(i-1)));
      V1(3)=pixel3;
      pixel4=(1+ M(j,(i-1)));
      V1(4)=pixel4;
      pixel5=(1.4142 + M((j+1),(i-1)));
      V1(5)=pixel5;
      pixel6=M(j,i);
      V1(6)=pixel6;
      minv=min(V1);
      M(j,i)= minv;
    end
 end
for j= ren-1: -1: 2,
   for i= col-1: -1: 2,
      pixel11=(1.4142 + M((j-1),(i+1)));
      V2(1)=pixel11;
      pixel12=(1 + M(j,(i+1)));
      V2(2)=pixel12;
      pixel13=(1.4142 + M((j+1),(i+1)));
      V2(3)=pixel13;
      pixel14=(1+ M((j+1),i));
      V2(4)=pixel14;
      pixel15=(1.4142 + M((j+1),(i-1)));
      V2(5)=pixel15;
      pixel16=M(j,i);
      V2(6)=pixel16;
      minva=min(V2);
      M(j,i)= minva;
    end
 end
 matriz1=M;
 objeto1= input('Nombre del objeto en la base de datos: ','s');
 fprintf('Se ha terminado la transformada de distancia de ')
 disp(objeto1)
 disp('Si se quiere ver la transformada de distancia, llamar a la variable matriz1')
```

entrenamiento2.m

```
%entrenamiento2

global matriz2
global objeto2

%el programa lee la imagen
I1=imread('foto84.jpg');
%se muestra la imagen original en una figura
%imshow(I1)
%title('Imagen Original')
%convertir la imagen a intensidad (grayscale)
I2=I1(:,:,1);
%procedimiento para obtener el contorno
%I3=wiener2(I2);
E=edge(I2,'canny');
E2=bwperim(E);

%procesamiento para cerrar el contorno
B2=bwmorph(E2, 'close');
B=bwfill(E2, 'holes');
B1=bwmorph(B, 'remove');
B3=bwmorph(B1, 'thicken');
%se muestran los contornos de las imagenes
figure,imshow(B3)
%title('Contornos de la Imagen')

%seleccion
BS=bwselect(B3, 8);
figure, imshow(BS)
title('Contorno seleccionado')

%matriz de distancias
M=zeros(size(BS)); %se crea una matriz del mismo tamaño que la matriz original
[r,c]=size(BS); %se obtiene el numero de renglones y columnas de la matriz original
%se hace un barrido que cambia los valores
for j=1: r,
   for i=1: c,
      pixel=BS(j,i);
      if (pixel==0)
         M(j,i)=255;
      else M(j,i)=0;
      end
  end
end


%transformada de distancia
[ren,col]=size(M);
V1=zeros(1,6);
V2=zeros(1,6);
for j=2: ren-1,
   for i=2: col-1,
      pixel1=(1.4142 + M((j-1),(i+1)));
       V1(1)=pixel1;
      pixel2=(1 + M((j-1),i));
```

```
        V1(2)=pixel2;
        pixel3=(1.4142 + M((j-1),(i-1)));
        V1(3)=pixel3;
        pixel4=(1+ M(j,(i-1)));
        V1(4)=pixel4;
        pixel5=(1.4142 + M((j+1),(i-1)));
        V1(5)=pixel5;
        pixel6=M(j,i);
        V1(6)=pixel6;
        minv=min(V1);
        M(j,i)= minv;
      end
  end
for j= ren-1: -1: 2,
    for i= col-1: -1: 2,
        pixel11=(1.4142 + M((j-1),(i+1)));
        V2(1)=pixel11;
        pixel12=(1 + M(j,(i+1)));
        V2(2)=pixel12;
        pixel13=(1.4142 + M((j+1),(i+1)));
        V2(3)=pixel13;
        pixel14=(1+ M((j+1),i));
        V2(4)=pixel14;
        pixel15=(1.4142 + M((j+1),(i-1)));
        V2(5)=pixel15;
        pixel16=M(j,i);
        V2(6)=pixel16;
        minva=min(V2);
        M(j,i)= minva;
      end
  end
  matriz2=M;
  objeto2= input('Nombre del objeto en la base de datos: ','s');
  fprintf('Se ha terminado la transformada de distancia de ')
  disp(objeto2)
  disp('Si se quiere ver la transformada de distancia, llamar a la variable matriz2')
```

entrenamiento3.m

```
%entrenamiento3
global matriz3
global objeto3

%el programa lee la imagen
I1=imread('foto97.jpg');
%se muestra la imagen original en una figura
%imshow(I1)
%title('Imagen Original')
%convertir la imagen a intensidad (grayscale)
I2=I1(:,:,1);
%procedimiento para obtener el contorno
%I3=wiener2(I2);
E=edge(I2,'canny');
E2=bwperim(E);
```

```
%procesamiento para cerrar el contorno
B2=bwmorph(E2, 'close');
B=bwfill(E2, 'holes');
B1=bwmorph(B, 'remove');
B3=bwmorph(B1, 'thicken');
%se muestran los contornos de las imagenes
figure,imshow(B3)
%title('Contornos de la Imagen')

%seleccion
BS=bwselect(B3, 8);
figure, imshow(BS)
title('Contorno seleccionado')

%matriz de distancias
M=zeros(size(BS)); %se crea una matriz del mismo tamaño que la matriz original
[r,c]=size(BS); %se obtiene el numero de renglones y columnas de la matriz original
%se hace un barrido que cambia los valores
for j=1: r,
   for i=1: c,
      pixel=BS(j,i);
      if (pixel==0)
        M(j,i)=255;
      else M(j,i)=0;
      end
  end
end

%transformada de distancia
[ren,col]=size(M);
V1=zeros(1,6);
V2=zeros(1,6);
for j=2: ren-1,
   for i=2: col-1,
      pixel1=(1.4142 + M((j-1),(i+1)));
      V1(1)=pixel1;
      pixel2=(1 + M((j-1),i));
      V1(2)=pixel2;
      pixel3=(1.4142 + M((j-1),(i-1)));
      V1(3)=pixel3;
      pixel4=(1+ M(j,(i-1)));
      V1(4)=pixel4;
      pixel5=(1.4142 + M((j+1),(i-1)));
      V1(5)=pixel5;
      pixel6=M(j,i);
      V1(6)=pixel6;
      minv=min(V1);
      M(j,i)= minv;
    end
 end
for j= ren-1: -1: 2,
   for i= col-1: -1: 2,
      pixel11=(1.4142 + M((j-1),(i+1)));
      V2(1)=pixel11;
      pixel12=(1 + M(j,(i+1)));
```

```
      V2(2)=pixel12;
      pixel13=(1.4142 + M((j+1),(i+1)));
      V2(3)=pixel13;
      pixel14=(1+ M((j+1),i));
      V2(4)=pixel14;
      pixel15=(1.4142 + M((j+1),(i-1)));
      V2(5)=pixel15;
      pixel16=M(j,i);
      V2(6)=pixel16;
      minva=min(V2);
      M(j,i)= minva;
    end
 end
 matriz3=M;
 objeto3= input('Nombre del objeto en la base de datos: ','s');
 fprintf('Se ha terminado la transformada de distancia de ')
 disp(objeto3)
 disp('Si se quiere ver la transformada de distancia, llamar a la variable matriz3')
```

entrenamiento4.m

```
%entrenamiento4
global matriz4
global objeto4

%el programa lee la imagen
I1=imread('foto67.jpg');
%se muestra la imagen original en una figura
%imshow(I1)
%title('Imagen Original')
%convertir la imagen a intensidad (grayscale)
I2=I1(:,:,1);
%procedimiento para obtener el contorno
%I3=wiener2(I2);
E=edge(I2,'canny');
E2=bwperim(E);

%procesamiento para cerrar el contorno
B2=bwmorph(E2, 'close');
B=bwfill(E2, 'holes');
B1=bwmorph(B, 'remove');
B3=bwmorph(B1, 'thicken');
%se muestran los contornos de las imagenes
figure,imshow(B3)
%title('Contornos de la Imagen')

%seleccion
BS=bwselect(B3, 8);
figure, imshow(BS)
title('Contorno seleccionado')

%matriz de distancias
M=zeros(size(BS)); %se crea una matriz del mismo tamaño que la matriz original
[r,c]=size(BS); %se obtiene el numero de renglones y columnas de la matriz original
%se hace un barrido que cambia los valores
```

```
for j=1: r,
   for i=1: c,
       pixel=BS(j,i);
       if (pixel==0)
          M(j,i)=255;
       else M(j,i)=0;
       end
  end
end


%transformada de distancia
[ren,col]=size(M);
V1=zeros(1,6);
V2=zeros(1,6);
for j=2: ren-1,
   for i=2: col-1,
       pixel1=(1.4142 + M((j-1),(i+1)));
       V1(1)=pixel1;
       pixel2=(1 + M((j-1),i));
       V1(2)=pixel2;
       pixel3=(1.4142 + M((j-1),(i-1)));
       V1(3)=pixel3;
       pixel4=(1+ M(j,(i-1)));
       V1(4)=pixel4;
       pixel5=(1.4142 + M((j+1),(i-1)));
       V1(5)=pixel5;
       pixel6=M(j,i);
       V1(6)=pixel6;
       minv=min(V1);
       M(j,i)= minv;
    end
 end

for j= ren-1: -1: 2,
   for i= col-1: -1: 2,
       pixel11=(1.4142 + M((j-1),(i+1)));
       V2(1)=pixel11;
       pixel12=(1 + M(j,(i+1)));
       V2(2)=pixel12;
       pixel13=(1.4142 + M((j+1),(i+1)));
       V2(3)=pixel13;
       pixel14=(1+ M((j+1),i));
       V2(4)=pixel14;
       pixel15=(1.4142 + M((j+1),(i-1)));
       V2(5)=pixel15;
       pixel16=M(j,i);
       V2(6)=pixel16;
       minva=min(V2);
       M(j,i)= minva;
    end
 end
 matriz4=M;
 objeto4= input('Nombre del objeto en la base de datos: ','s');
 fprintf('Se ha terminado la transformada de distancia de ')
```

```
 disp(objeto4)
 disp('Si se quiere ver la transformada de distancia, llamar a la variable matriz4')
```

reconocimiento.m

```
%prueba o comparacion final

function [prom]= reconocimiento(I)

procesamiento2(I)
seleccion
mapeo
```

procesamiento2.m

```
%procesamiento de imagen y contorno del objeto
%este programa obtiene el contorno de una lata:
%lee la imagen, la convierte a intensidad, obtiene el contorno
%aplicando el metodo canny y una convolucion
%finalmente se vuelve a procesar la imagen para obtener un contorno cerrado


function [] = procesamiento2(I)

global B3;
%el programa lee la imagen
I1=imread(I);
%se muestra la imagen original en una figura
%imshow(I1)
%title('Imagen Original')
%convertir la imagen a intensidad (grayscale)
I2=I1(:,:,1);
%figure, imshow(I2)
%procedimiento para obtener el contorno
%I3=wiener2(I2);
E=edge(I2,'canny');
%figure, imshow(E)
E2=bwperim(E);
%figure, imshow(E2)

%procesamiento para cerrar el contorno
B2=bwmorph(E2, 'close');
B=bwfill(B2, 'holes');
B1=bwmorph(B, 'remove');
B3=bwmorph(B1, 'thicken');
%se muestran los contornos de las imagenes
%figure,imshow(B3)
%title('Contornos de la Imagen')
```

selección.m

```
%seleccion del contorno de interes
%se utiliza para seleccionar un contorno
%de todos los que se encuentran en la imagen
%con el boton izquierdo se marcan de 1 a 8 puntos
%en el contorno de interes y un click derecho marca
```

```
%el punto final


global B3

BS=bwselect(B3, 8);
figure, imshow(BS)
title('Contorno seleccionado')
```

mapeo.m

```
%mapeo o comparacion
%este comando realiza la comparacion del contorno de una figura cualquiera
%con las matrices de distancia que se tienen como base de datos
global matriz1
global objeto1
global matriz2
global objeto2
global matriz3
global objeto3
global matriz4
global objeto4

Vprom=zeros(1,4);

%comparacion con el primer objeto
[r,c]=size(BS);
suma=0;
k=0;
for j=1: r,
   for i=1: c,
       elem=BS(j,i);
       if (elem==1)
          suma= suma + matriz1(j,i);
          k= k+1;

       end
  end
end

prom1=suma/k;

fprintf('La distancia del objeto con respecto al primer objeto = ')
disp(prom1)
%se utiliza cuando el objeto conocido y el reconocido son identicos
if(prom1==0)
   fprintf('El objeto es un(a) ')
   disp(objeto1)
end

%comparacion con  el segundo objeto
[r,c]=size(BS);
suma=0;
k=0;
for j=1: r,
```

```
   for i=1: c,
      elem=BS(j,i);
      if (elem==1)
        suma= suma + matriz2(j,i);
        k= k+1;
      end
  end
end

prom2=suma/k;

fprintf('La distancia del objeto con respecto al segundo objeto = ')
disp(prom2)
%se utiliza cuando el objeto conocido y el reconocido son identicos
if(prom2==0)
   fprintf('El objeto es un(a) ')
   disp(objeto2)
end

%comparacion con el tercer objeto
[r,c]=size(BS);
suma=0;
k=0;
for j=1: r,
   for i=1: c,
      elem=BS(j,i);
      if (elem==1)
        suma= suma + matriz3(j,i);
        k= k+1;
      end
  end
end

prom3=suma/k;

fprintf('La distancia del objeto con respecto al tercer objeto = ')
disp(prom3)
%se utiliza cuando el objeto conocido y el reconocido son identicos
if(prom3==0)
   fprintf('El objeto es un(a) ')
   disp(objeto3)
end

%comparacion con el cuarto objeto
[r,c]=size(BS);
suma=0;
k=0;
for j=1: r,
   for i=1: c,
      elem=BS(j,i);
      if (elem==1)
        suma= suma + matriz4(j,i);
        k= k+1;
      end
  end
```

```
end

prom4=suma/k;

fprintf('La distancia del objeto con respecto al cuarto objeto = ')
disp(prom4)
%se utiliza cuando el objeto conocido y el reconocido son identicos
if(prom4==0)
  fprintf('El objeto es un(a) ')
          disp(objeto4)
end

Vprom(1)=prom1;
Vprom(2)=prom2;
Vprom(3)=prom3;
Vprom(4)=prom4;

valormin= min(Vprom);


switch (valormin)
        case (valormin==prom1)
                 fprintf('El objeto es un(a) ')
                 disp(objeto1)
        case (valormin==prom2)
                 fprintf('El objeto es un(a) ')
                 disp(objeto2)
        case (valormin==prom3)
                 fprintf('El objeto es un(a) ')
                 disp(objeto3)
        otherwise
                 fprintf('El objeto es un(a) ')
                 disp(objeto4)
        end
      end
   end
end
```