Universidad de Santiago de Chile
Facultad de Ingeniería
Departamento de Ingeniería Eléctrica

Visión Artificial

Tarea N°3

"Reconocimiento de Placa patente".

Alumno: Guillermo Echagüe Arriaza.
Profesor: Renato Salinas Silva.
Curso: Visión Artificial.
Fecha: 23-09-2015.

Contenido

Introducción

El reconocimiento de objetos es la última etapa dentro de un sistema de visión por computador. A partir de las características encontradas y de los posibles objetos que el conocimiento a priori del problema se espera que puedan aparecer, el sistema debe determinar que objetos están presentes en la imagen. Esta etapa es la de mayor grado de abstracción de todas las que forman un sistema de visión y a menudo se realizan fuertes simplificaciones para que funcione con éxito.

En este informe se realizara la confección de un software que permita determinar y encontrar la ubicación de los caracteres de una placa patente Chilena, identificándolos y determinando el valor de ese carácter.

Además para la detección de ruidos se utilizaran algoritmos de erosión y dilatación, para la eliminación de estos, también se debe diseñar un clasificador que reconozca el objeto a la vista de la información extraída de la imagen y los posibles candidatos.

Objetivo

Capturar imágenes con una webcam de una placa patente chilena y desplegarlas en una ventana del computador, utilizando software abierto como Matlab, Labview u OpenCV. El sistema deberá localizar la patente (dentro de una región acotada) y luego eliminar o ignorar los ruidos, determinando los caracteres de la placa. Para este proyecto el ángulo y la distancia deben ser fijos (con alguna tolerancia).

Descripción de la actividad

El sistema debe poder mostrar en una ventana el video (en vivo), en otra ventana la imagen capturada y luego desplegar los valores estimados de los caracteres.

El sistema debe funcionar con patentes reales en ambientes físicos. Por lo tanto, se pide capturar 10 imágenes de patentes distintas por grupo. Dichas imágenes deben venir en el CD o DVD que acompaña su Informe. Deberá adjuntar a dichas imágenes un texto que describa la cámara usada, la resolución, la distancia y la orientación de forma adecuada. (Incluya distancia focal si la conoce). Trate de capturar video de algunos segundos de cada patente (la foto es obligatoria).

Desarrollo

Se usará la capacidad de una webcam para el procesamiento de imágenes bajo control de un programa como MatLab.

El sistema debe poder mostrar además en una ventana el video de la captura en línea, en otra ventana la imagen capturada, y en otra la imagen segmentada. Para mayor flexibilidad el dato de distancia podría ser ingresado por el usuario. Para la demostración en clases se sugiere usar un cartón blanco con caracteres y logos como los de una patente real con un borde negro. (Puede ampliar una foto al tamaño adecuado). El proyecto también incluye el análisis de imágenes capturadas en exteriores de varias placas patentes.

Materiales

WEBCAM

Shark Net SN CINEMA II	
Especificaciones de desempeño	
Megapixels	5M Pixel
Resolución de video e imagen	2560 x 1920 MAX (6 FPS)
Foco Ajustable	Si
Micrófono	Si, 5mts de alcance
Formato de video	24Bits
Angulo de rotación	Base giratoria de 360º. Visión nocturna
Formato de almacenamiento	Compresión de imagen.

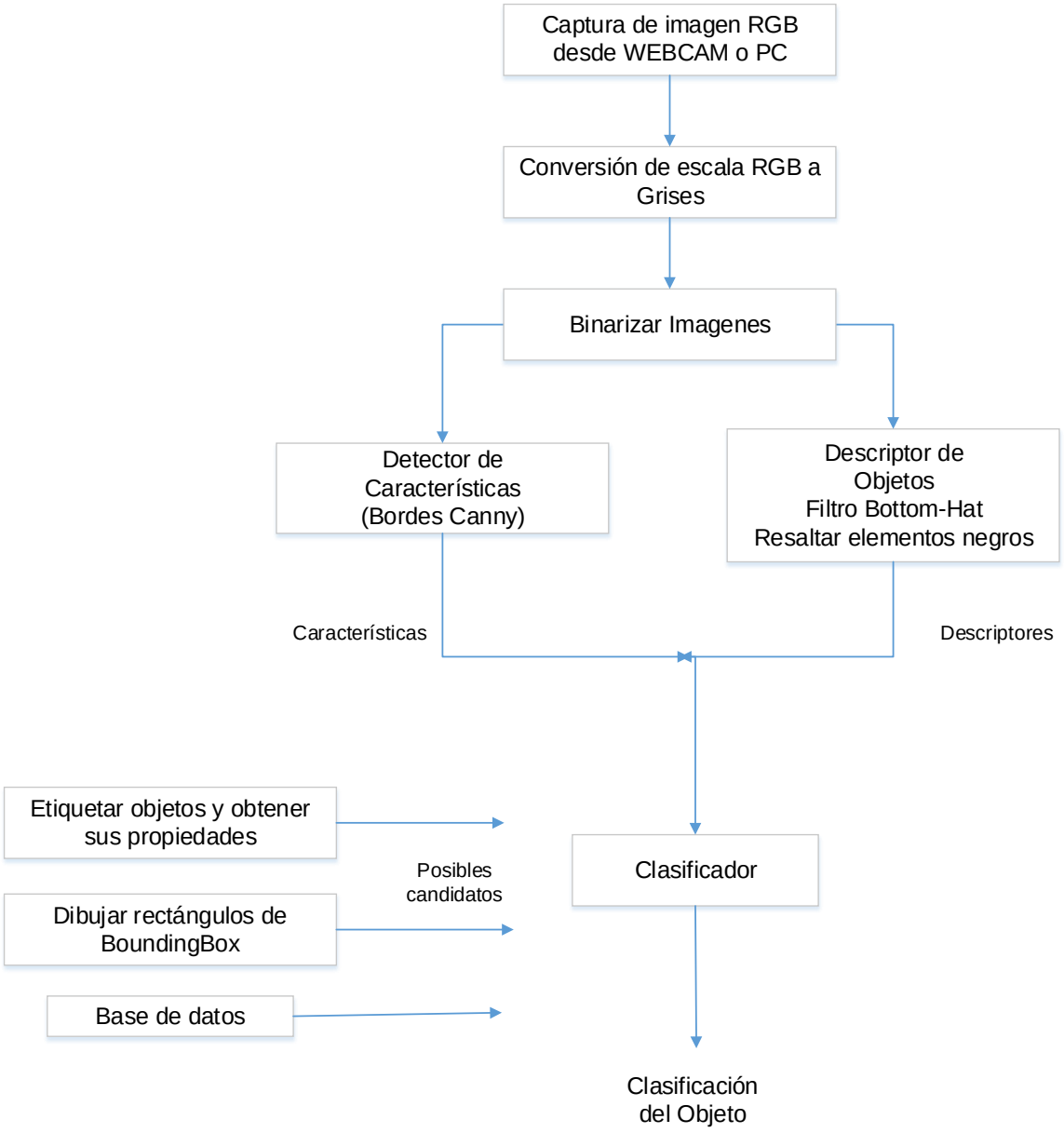
Patentes Chilenas

Métodos

En el reconocimiento de objetos a partir de las características encontradas y de los posibles objetos que el conocimiento a priori del problema se espera que puedan aparecer, las características deseadas y el sistema debe determinar qué objetos están presentes en la imagen.

A partir de la imagen original o procesada se obtienen una serie de características (color, textura, etc.) y/o descriptores (momentos, descriptores de Fourier, etc.) que definen cada objeto. Con ello se ha pretendido disminuir el volumen de la información hasta hacerla manejable pero sin perder ninguna información vital o valiosa. La representación de los objetos por medio de estas características se habrá realizado mediante el análisis previo de otras imágenes en donde se habrá comprobado que realmente definen a los posibles objetos y los hacen distinguibles unos de otro.

Diagrama de Flujo



Descripción de Algoritmos

Capturar Videos desde WEBCAM

```
% --- Executes on button press in cap_video.
function cap_video_Callback(hObject, eventdata, handles)
global vid
vid=videoinput('winvideo',1,'YUY2_320x240');
% Captura una trama por trigger
vid.FramesPerTrigger=1;
% Salida de la Imagen RGB=color/GRAYSCALE=escala grises
vid.ReturnedColorspace='grayscale';
% Se le indica a Matlab que la cámara no inicie Automáticamente sino a
% petición del usuario
riggerconfig(vid,'manual');
% Adquirimos la altura y anchura de la imagen
vidRes=get(vid,'VideoResolution');
% Altura de la Imagen
imWidth = vidRes(1);
% Anchura de la Imagen
imHeight = vidRes(2);
nBands = get(vid, 'NumberOfBands');
% Crea una Variable q contenga la Imagen para mostrarla en el AXIS(handles.imagen_1)
hImage = image(zeros(imHeight, imWidth, nBands), 'parent', handles.imagen_1);
% Empieza Webcam Preview
preview(vid, hImage);
```

Capturar Imagen desde video de WEBCAM

```
% --- Executes on button press in cap_imagen.
function cap_imagen_Callback(hObject, eventdata, handles)
global vid Img
Img = getsnapshot(vid);
axes(handles.imagen_1);
BW = edge(Img,'canny');
imshow(Img);
pause (0.1);
```

Cargar imagen * jpg desde computador

```
% --- Executes on button press in car_imagen.
function car_imagen_Callback(hObject, eventdata, handles)
global im I
filename = uigetfile('*.jpg','Select an image file');
im = imread(filename);
% Datos de la escala y la pantalla como una imagen
imagesc(im);
% Mostar imagen en el axe.
axes(handles.imagen_2);
% Convertir a Gris
I=rgb2gray(im);
% Mostrar imagen
imshow(I);
pause (0.1);
```

```
%% Normalizar Contraste
X=Img;
[N,M] = size(X);
    Y = zeros(N,M);
    [i,j] = sort(X(:));
    z = zeros(N*M,1);
    d = fix(N*M/256+0.5);
    for i=1:255;
        z((i-1)*d+1:i*d) = (i-1)*ones(d,1);
    end
    z(255*d+1:N*M) = 255*ones(N*M-255*d,1);
    Y(j) = z;
im_g=X;

%% Binarización
umb=graythresh(Img);
bw=imcomplement(im2bw(Img,umb));
bw=medfilt2(bw, [8, 8], 'symmetric');
handles.bw=bw;

%% Etiquetar elementos conectados
[L Ne]=bwlabel(bw);
handles.Ne=Ne;

%% Calcular propiedades de los objetos de la imagen
propied= regionprops(L);
handles.propied=propied;
hold on;
```

Creación de un OCR para el reconocimiento de caracteres

```
%% Carga de caracteres numeros para el OCR
caracter0=(imread('caracteres\0.bmp'));
caracter1=(imread('caracteres\1.bmp'));
caracter2=(imread('caracteres\2.bmp'));
caracter3=(imread('caracteres\3.bmp'));
caracter4=(imread('caracteres\4.bmp'));
caracter5=(imread('caracteres\5.bmp'));
caracter6=(imread('caracteres\6.bmp'));
caracter7=(imread('caracteres\7.bmp'));
caracter8=(imread('caracteres\8.bmp'));
caracter9=(imread('caracteres\9.bmp'));

%% Carga de caracteres numeros para el OCR
caracterA=(imread('caracteres\A.bmp'));
caracterB=(imread('caracteres\B.bmp'));
caracterC=(imread('caracteres\C.bmp'));
caracterD=(imread('caracteres\D.bmp'));
caracterE=(imread('caracteres\E.bmp'));
caracterF=(imread('caracteres\F.bmp'));
caracterG=(imread('caracteres\G.bmp'));
caracterH=(imread('caracteres\H.bmp'));
caracterI=(imread('caracteres\I.bmp'));
caracterJ=(imread('caracteres\J.bmp'));
caracterK=(imread('caracteres\K.bmp'));
```

```
characterL=(imread('caracteres\L.bmp'));
characterM=(imread('caracteres\M.bmp'));
characterN=(imread('caracteres\N.bmp'));
characterO=(imread('caracteres\O.bmp'));
characterP=(imread('caracteres\P.bmp'));
characterQ=(imread('caracteres\Q.bmp'));
characterR=(imread('caracteres\R.bmp'));
characterS=(imread('caracteres\S.bmp'));
characterT=(imread('caracteres\T.bmp'));
characterU=(imread('caracteres\U.bmp'));
characterV=(imread('caracteres\V.bmp'));
characterW=(imread('caracteres\W.bmp'));
characterX=(imread('caracteres\X.bmp'));
characterY=(imread('caracteres\Y.bmp'));
characterZ=(imread('caracteres\Z.bmp'));

%% Redimensionado de caracteres número
caracteres{1,1}=imresize(caracter0,[68 40]);
caracteres{2,1}=imresize(caracter1,[68 40]);
caracteres{3,1}=imresize(caracter2,[68 40]);
caracteres{4,1}=imresize(caracter3,[68 40]);
caracteres{5,1}=imresize(caracter4,[68 40]);
caracteres{6,1}=imresize(caracter5,[68 40]);
caracteres{7,1}=imresize(caracter6,[68 40]);
caracteres{8,1}=imresize(caracter7,[68 40]);
caracteres{9,1}=imresize(caracter8,[68 40]);
caracteres{10,1}=imresize(caracter9,[68 40]);

%% Redimensionado de caracteres letras
caracteres{11,1}=imresize(caracterA,[68 40]);
caracteres{12,1}=imresize(caracterB,[68 40]);
caracteres{13,1}=imresize(caracterC,[68 40]);
caracteres{14,1}=imresize(caracterD,[68 40]);
caracteres{15,1}=imresize(caracterE,[68 40]);
caracteres{16,1}=imresize(caracterF,[68 40]);
caracteres{17,1}=imresize(caracterG,[68 40]);
caracteres{18,1}=imresize(caracterH,[68 40]);
caracteres{19,1}=imresize(caracterI,[68 40]);
caracteres{20,1}=imresize(caracterJ,[68 40]);
caracteres{21,1}=imresize(caracterK,[68 40]);
caracteres{22,1}=imresize(caracterL,[68 40]);
caracteres{23,1}=imresize(caracterM,[68 40]);
caracteres{24,1}=imresize(caracterN,[68 40]);
caracteres{25,1}=imresize(caracterO,[68 40]);
caracteres{26,1}=imresize(caracterP,[68 40]);
caracteres{27,1}=imresize(caracterQ,[68 40]);
caracteres{28,1}=imresize(caracterR,[68 40]);
caracteres{29,1}=imresize(caracterS,[68 40]);
caracteres{30,1}=imresize(caracterT,[68 40]);
caracteres{31,1}=imresize(caracterU,[68 40]);
caracteres{32,1}=imresize(caracterV,[68 40]);
caracteres{33,1}=imresize(caracterW,[68 40]);
caracteres{34,1}=imresize(caracterX,[68 40]);
caracteres{35,1}=imresize(caracterY,[68 40]);
caracteres{36,1}=imresize(caracterZ,[68 40]);
```

```
%% Asignacion de caracteres para ocr
ocr(1,1)='0';
ocr(2,1)='1';
ocr(3,1)='2';
ocr(4,1)='3';
ocr(5,1)='4';
ocr(6,1)='5';
ocr(7,1)='6';
ocr(8,1)='7';
ocr(9,1)='8';
ocr(10,1)='9';

ocr(11,1)='A';
ocr(12,1)='B';
ocr(13,1)='C';
ocr(14,1)='D';
ocr(15,1)='E';
ocr(16,1)='F';
ocr(17,1)='G';
ocr(18,1)='H';
ocr(19,1)='I';
ocr(20,1)='J';
ocr(21,1)='K';
ocr(22,1)='L';
ocr(23,1)='M';
ocr(24,1)='N';
ocr(25,1)='O';
ocr(26,1)='P';
ocr(27,1)='Q';
ocr(28,1)='R';
ocr(29,1)='S';
ocr(30,1)='T';
ocr(31,1)='U';
ocr(32,1)='V';
ocr(33,1)='W';
ocr(34,1)='X';
ocr(35,1)='Y';
ocr(36,1)='Z';

handles.caracteres=caracteres;
handles.ocr=ocr;
```

Pruebas

Pruebas realizadas por medio de la WEBCAM

Figura 1

Figura 2

Pruebas obtenidas por medio de imágenes

Figura 3

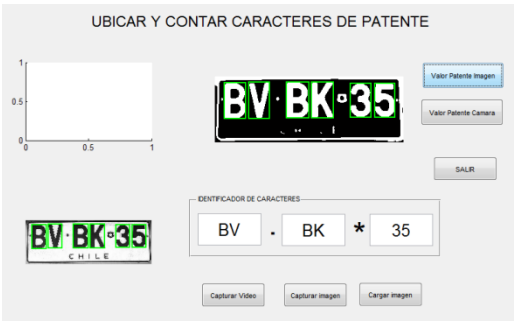


Figura 4

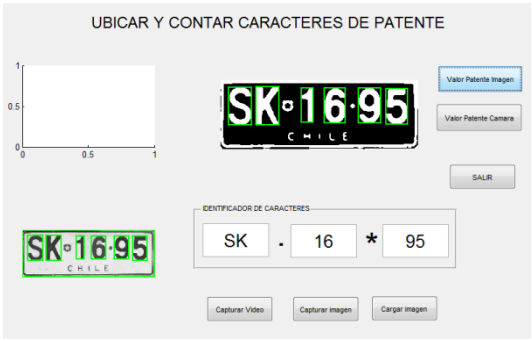


Figura 5

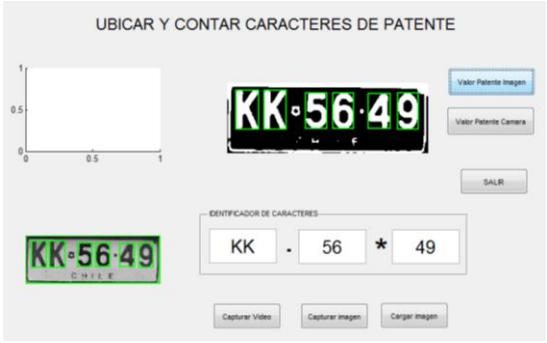


Figura 6

Figura 7

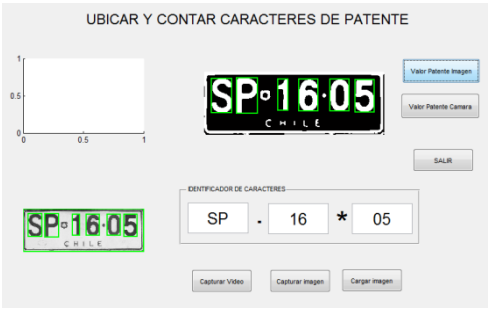


Figura 8

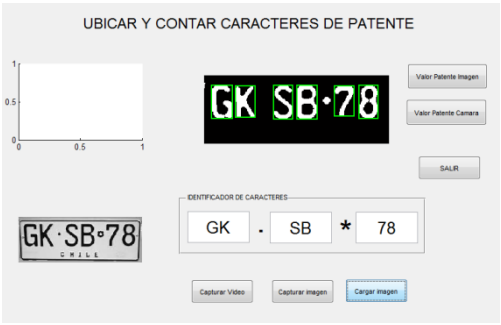


Figura 9

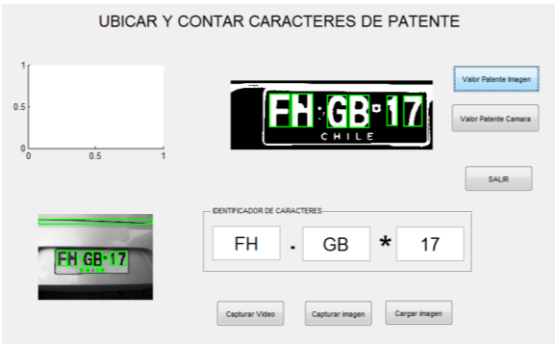


Figura 10

Figura 11

Conclusiones

Para determinar el valor de los caracteres por parte del programa, se debe realizar un análisis por parte del operador humano de las imágenes de prueba obtenidas de una base de datos que contiene los modelos de los objetos que puedan aparecer en la imagen. El contenido de esta base de datos irá muy ligado al tipo de características que se van a obtener de la imagen, en este trabajo será principalmente las letras y números, para determinar las combinaciones de las placas patente.

Para acelerar el análisis de las imágenes, se debe eliminar estructuras y objetos irrelevantes al objetivo propuesto, pero sin perder información importante de la imagen. También se deben considerar los elementos conectados, en los cuales se realizaran las operaciones Morfológicas como Erosión y Dilatación, para facilitar la obtención de la información requerida del análisis de las imágenes.

Los resultados son variables de acuerdo a la cantidad y tipo de luz presente en el ambiente del análisis lo que hace variar el tipo de umbral a utilizar, ya que este marca la principal diferencia a la hora de detectar los caracteres, sin perder información importante.

Finalmente dentro de las características más importantes para este trabajo está el generar una imagen patrón que tenga relación con las características de luz y contraste para realizar la identificación y reconocimiento del carácter de la placa patente, mencionado anteriormente.

Bibliografía

[1] De la Escalera, Arturo. Visión Por computadora: Fundamentos y Métodos. Madrid: Prentice Hall (2001)

[2] Image Processing Toolbox™ User's Guide (2015). Mathworks

Apéndice

Código en MatLab

```
function varargout = Tarea_3_Vision(varargin)

gui_Singleton = 1;
gui_State = struct('gui_Name',       mfilename, ...
                   'gui_Singleton',  gui_Singleton, ...
                   'gui_OpeningFcn', @Tarea_2_Vision_OpeningFcn, ...
                   'gui_OutputFcn',  @Tarea_2_Vision_OutputFcn, ...
                   'gui_LayoutFcn',  [] , ...
                   'gui_Callback',   []);
if nargin && ischar(varargin{1})
    gui_State.gui_Callback = str2func(varargin{1});
end

if nargout
    [varargout{1:nargout}] = gui_mainfcn(gui_State, varargin{:});
else
    gui_mainfcn(gui_State, varargin{:});
end
% End initialization code - DO NOT EDIT

% --- Executes just before Tarea_3_Vision is made visible.
function Tarea_3_Vision_OpeningFcn(hObject, eventdata, handles, varargin)

handles.output = hObject;

% Update handles structure
guidata(hObject, handles);

% UIWAIT makes Tarea_3_Vision wait for user response (see UIRESUME)
% uiwait(handles.figure1);

% --- Outputs from this function are returned to the command line.
function varargout = Tarea_3_Vision_OutputFcn(hObject, eventdata, handles)

varargout{1} = handles.output;

% --- Executes on button press in salir.
function salir_Callback(hObject, eventdata, handles)
close();

function valor_1_Callback(hObject, eventdata, handles)

% --- Executes during object creation, after setting all properties.
function valor_1_CreateFcn(hObject, eventdata, handles)

if ispc && isequal(get(hObject,'BackgroundColor'), get(0,'defaultUicontrolBackgroundColor'))
    set(hObject,'BackgroundColor','white');
end

function valor_2_Callback(hObject, eventdata, handles)
```

```
% --- Executes during object creation, after setting all properties.
function valor_2_CreateFcn(hObject, eventdata, handles)

if ispc && isequal(get(hObject,'BackgroundColor'), get(0,'defaultUicontrolBackgroundColor'))
    set(hObject,'BackgroundColor','white');
end

function valor_3_Callback(hObject, eventdata, handles)

% --- Executes during object creation, after setting all properties.
function valor_3_CreateFcn(hObject, eventdata, handles)

if ispc && isequal(get(hObject,'BackgroundColor'), get(0,'defaultUicontrolBackgroundColor'))
    set(hObject,'BackgroundColor','white');
end

% --- Executes on button press in video.
function video_Callback(hObject, eventdata, handles)
global vid
vid=videoinput('winvideo',1,'YUY2_320x240');
% Captura una trama por trigger
vid.FramesPerTrigger=1;
% Salida de la Imagen RGB=color/GRAYSCALE=escala grises
vid.ReturnedColorspace='grayscale';
% Se le indica a Matlab que la camara no inicie Automaticamente sino a
% peticion del usuario
triggerconfig(vid,'manual');
% Adquirimos la altura y anchura de la imagen
vidRes=get(vid,'VideoResolution');
% Altura de la Imagen
imWidth = vidRes(1);
% Anchura de la Imagen
imHeight = vidRes(2);
nBands = get(vid, 'NumberOfBands');
% Crea una Variable q contenga la Imagen para mostrala en el AXIS(handles.axes1)
hImage = image(zeros(imHeight, imWidth, nBands), 'parent', handles.axes1);
% Empieza Webcam Preview
preview(vid, hImage);

% --- Executes on button press in imagen.
function imagen_Callback(hObject, eventdata, handles)
global vid Img
Img = getsnapshot(vid);
axes(handles.axes1);
BW = edge(Img,'canny');
imshow(Img);
pause (0.1);
%% Mostrar imagen
imshow(Img);
pause (0.1);

%% Carga de caracteres numeros para el OCR
caracter0=(imread('caracteres\0.bmp'));
```

```
caracter1=(imread('caracteres\1.bmp'));
caracter2=(imread('caracteres\2.bmp'));
caracter3=(imread('caracteres\3.bmp'));
caracter4=(imread('caracteres\4.bmp'));
caracter5=(imread('caracteres\5.bmp'));
caracter6=(imread('caracteres\6.bmp'));
caracter7=(imread('caracteres\7.bmp'));
caracter8=(imread('caracteres\8.bmp'));
caracter9=(imread('caracteres\9.bmp'));

%% Carga de caracteres numeros para el OCR
caracterA=(imread('caracteres\A.bmp'));
caracterB=(imread('caracteres\B.bmp'));
caracterC=(imread('caracteres\C.bmp'));
caracterD=(imread('caracteres\D.bmp'));
caracterE=(imread('caracteres\E.bmp'));
caracterF=(imread('caracteres\F.bmp'));
caracterG=(imread('caracteres\G.bmp'));
caracterH=(imread('caracteres\H.bmp'));
caracterI=(imread('caracteres\I.bmp'));
caracterJ=(imread('caracteres\J.bmp'));
caracterK=(imread('caracteres\K.bmp'));
caracterL=(imread('caracteres\L.bmp'));
caracterM=(imread('caracteres\M.bmp'));
caracterN=(imread('caracteres\N.bmp'));
caracterO=(imread('caracteres\O.bmp'));
caracterP=(imread('caracteres\P.bmp'));
caracterQ=(imread('caracteres\Q.bmp'));
caracterR=(imread('caracteres\R.bmp'));
caracterS=(imread('caracteres\S.bmp'));
caracterT=(imread('caracteres\T.bmp'));
caracterU=(imread('caracteres\U.bmp'));
caracterV=(imread('caracteres\V.bmp'));
caracterW=(imread('caracteres\W.bmp'));
caracterX=(imread('caracteres\X.bmp'));
caracterY=(imread('caracteres\Y.bmp'));
caracterZ=(imread('caracteres\Z.bmp'));

%% Redimensionado de caracteres número
caracteres{1,1}=imresize(caracter0,[68 40]);
caracteres{2,1}=imresize(caracter1,[68 40]);
caracteres{3,1}=imresize(caracter2,[68 40]);
caracteres{4,1}=imresize(caracter3,[68 40]);
caracteres{5,1}=imresize(caracter4,[68 40]);
caracteres{6,1}=imresize(caracter5,[68 40]);
caracteres{7,1}=imresize(caracter6,[68 40]);
caracteres{8,1}=imresize(caracter7,[68 40]);
caracteres{9,1}=imresize(caracter8,[68 40]);
caracteres{10,1}=imresize(caracter9,[68 40]);

%% Redimensionado de caracteres letras
caracteres{11,1}=imresize(caracterA,[68 40]);
caracteres{12,1}=imresize(caracterB,[68 40]);
caracteres{13,1}=imresize(caracterC,[68 40]);
caracteres{14,1}=imresize(caracterD,[68 40]);
```

```
caracteres{15,1}=imresize(caracterE,[68 40]);
caracteres{16,1}=imresize(caracterF,[68 40]);
caracteres{17,1}=imresize(caracterG,[68 40]);
caracteres{18,1}=imresize(caracterH,[68 40]);
caracteres{19,1}=imresize(caracterI,[68 40]);
caracteres{20,1}=imresize(caracterJ,[68 40]);
caracteres{21,1}=imresize(caracterK,[68 40]);
caracteres{22,1}=imresize(caracterL,[68 40]);
caracteres{23,1}=imresize(caracterM,[68 40]);
caracteres{24,1}=imresize(caracterN,[68 40]);
caracteres{25,1}=imresize(caracterO,[68 40]);
caracteres{26,1}=imresize(caracterP,[68 40]);
caracteres{27,1}=imresize(caracterQ,[68 40]);
caracteres{28,1}=imresize(caracterR,[68 40]);
caracteres{29,1}=imresize(caracterS,[68 40]);
caracteres{30,1}=imresize(caracterT,[68 40]);
caracteres{31,1}=imresize(caracterU,[68 40]);
caracteres{32,1}=imresize(caracterV,[68 40]);
caracteres{33,1}=imresize(caracterW,[68 40]);
caracteres{34,1}=imresize(caracterX,[68 40]);
caracteres{35,1}=imresize(caracterY,[68 40]);
caracteres{36,1}=imresize(caracterZ,[68 40]);

%% Asignacion de caracteres para ocr
ocr(1,1)='0';
ocr(2,1)='1';
ocr(3,1)='2';
ocr(4,1)='3';
ocr(5,1)='4';
ocr(6,1)='5';
ocr(7,1)='6';
ocr(8,1)='7';
ocr(9,1)='8';
ocr(10,1)='9';

ocr(11,1)='A';
ocr(12,1)='B';
ocr(13,1)='C';
ocr(14,1)='D';
ocr(15,1)='E';
ocr(16,1)='F';
ocr(17,1)='G';
ocr(18,1)='H';
ocr(19,1)='I';
ocr(20,1)='J';
ocr(21,1)='K';
ocr(22,1)='L';
ocr(23,1)='M';
ocr(24,1)='N';
ocr(25,1)='O';
ocr(26,1)='P';
ocr(27,1)='Q';
ocr(28,1)='R';
ocr(29,1)='S';
ocr(30,1)='T';
ocr(31,1)='U';
```

```
ocr(32,1)='V';
ocr(33,1)='W';
ocr(34,1)='X';
ocr(35,1)='Y';
ocr(36,1)='Z';

handles.caracteres=caracteres;
handles.ocr=ocr;

%% Fin Carga de caracteres

%% Normalizar Contraste
X=Img;
[N,M] = size(X);
   Y = zeros(N,M);
   [i,j] = sort(X(:));
   z = zeros(N*M,1);
   d = fix(N*M/256+0.5);
   for i=1:255;
      z((i-1)*d+1:i*d) = (i-1)*ones(d,1);
   end
   z(255*d+1:N*M) = 255*ones(N*M-255*d,1);
   Y(j) = z;
   im_g=X;

%% Binarización
umb=graythresh(Img);
bw=imcomplement(im2bw(Img,umb));
bw=medfilt2(bw, [8, 8], 'symmetric');
handles.bw=bw;

%% Etiquetar elementos conectados
[L Ne]=bwlabel(bw);
handles.Ne=Ne;

%% Calcular propiedades de los objetos de la imagen
propied= regionprops(L);
handles.propied=propied;
hold on;
propied.Area
%% Graficar las 'cajas' de frontera de los objetos
for n=1:size(propied,1)
   if propied(n).Area > 500 && propied(n).Area < 1500
   rectangle('Position',propied(n).BoundingBox,...
      'EdgeColor','g','LineWidth',2)
   pause (0.01)
   end
end
pause (0.1)
hold off;
guidata(hObject,handles);

% --- Executes on button press in patente.
function patente_Callback(hObject, eventdata, handles)
%% Recorte patente de la imagen
```

```
bw=imcomplement(handles.bw);

Ic=bw;
st=strel('disk',50);                %Elemento estructurante de 10 pixeles de radio
IM2=imbothat(Ic,st);                %Hace Bottom-Hat
I3=IM2;                             %Aplica Umbral
LH=strel('line',100,0);             %Elemento estructurante lineal horizontal
IM3=imclose(I3,LH);                 %Closing con elemento estructurante
LV=strel('line',20,140);            %Elemento estructurante lineal vertical
IM4=imopen(IM3,LV);                 %Hace opening con elemento estructurante
DIV=strel('line',70,90);            %Elemento estructurante lineal vertical
DIH=strel('line',40,0);             %Elemento estrucutrante lineal horizontal
IM5=imdilate(IM4,DIV);              %Dilata con E.E. vertical
IM6=imdilate(IM5,DIH);              %Dilata con E.E. horizontal
bw=IM6;
% figure
% imshow(bw);
[L Ne]=bwlabel(bw);
propied=regionprops(L);
clear boundingbox;
for i=1:Ne
    boundingbox(i,:)=propied(i,:).BoundingBox;
end
proporcion=boundingbox(:,3)./boundingbox(:,4);
puntero=find(min(abs(proporcion-3.5))==abs(proporcion-3.5));
boxplaca=propied(puntero,:).BoundingBox;    %Tamaño inicial de la region

boxplaca=[...
    boxplaca(1,1)...
    boxplaca(1,2)...
    boxplaca(1,3)...
    boxplaca(1,4)];
handles.placa=imcrop(handles.bw,boxplaca);      %Realiza el corte
axes(handles.axes3);
imshow(handles.placa);                  %Muestra imagen de la zona de la placa
guidata(hObject,handles);

%% Recortar Caracteres
placa=handles.placa;
caracteres=handles.caracteres;
ocr=handles.ocr;
[L Ne]=bwlabel(placa);
handles.Ne=Ne;
disp(Ne);

%% Calcular propiedades de los objetos de la imagen
propied= regionprops(L)
handles.propied=propied;
hold on;
propied.Area
%% Graficar las 'cajas' de frontera de los objetos
n=size(propied,1)

if n <= 12
```

```
for n=1:size(propied,1)
   if propied(n).Area > 300 && propied(n).Area < 10000
   rectangle('Position',propied(n).BoundingBox,...
      'EdgeColor','g','LineWidth',2)
   pause (0.01)
   end
end
end

if n <=20

for n=1:size(propied,1)
   if propied(n).Area > 1200 && propied(n).Area < 10000
   rectangle('Position',propied(n).BoundingBox,...
      'EdgeColor','g','LineWidth',2)
   pause (0.01)
   end
end
end

if n>20

for n=1:size(propied,1)
   if propied(n).Area > 25000 && propied(n).Area < 110000
   rectangle('Position',propied(n).BoundingBox,...
      'EdgeColor','g','LineWidth',2)
   pause (0.01)

   end
 end
end

pause (0.1)
hold off;
guidata(hObject,handles);
clear boundingbox;
for i=1:Ne
   boundingbox(i,:)=propied(i,:).BoundingBox;

end
proporcion=boundingbox(:,3)./boundingbox(:,4);
disp(proporcion);
puntero=find(proporcion>0.82 | proporcion<0.2);
proporcion(puntero)=0;
puntero=find(proporcion~=0);
[f c]=size(puntero);
disp(puntero);
caracter={};
for i=1:f
   boxplaca=propied(puntero(i,1),:).BoundingBox;
   caracter{i}=imcrop(placa,boxplaca);
end
disp(caracter);
[f c]=size(caracter);
sizes=[];
```

```
for i=1:c
    sizes=[sizes; size(caracter{i})];
end
disp(sizes);
[orden indice]=sort(sizes,1,'descend');
indice2=indice(1:6,1)
disp(indice2);
indice2=sort(indice2);
disp(indice2(:,1));
caracte2={};
for i=1:6
    caracter2{i}=caracter{indice2(i,1)};
end
disp(caracter2);

%% Escalar caracteres al tamaño del patrón o template
for i=1:6
    caracter_r{i}=imresize(caracter2{i},[68 40]);
end

%% Reconocimiento de caracteres
[f c]=size(caracter_r);
comp=zeros(15,1);
puntero=0;
patente=[''];

comp=0;
for i=1:2
    for j=11:36
        comp(j,1)=corr2(caracteres{j,1},caracter_r{i});
    end
    puntero=find(max(comp)==comp);
    patente=[patente ocr(puntero,1)];
end

comp=0;
for i=3:4
    for j=1:36
        comp(j,1)=corr2(caracteres{j,1},caracter_r{i});
    end
    puntero=find(max(comp)==comp);
    patente=[patente ocr(puntero,1)];
end

comp=0;
for i=5:6
    for j=1:10
        comp(j,1)=corr2(caracteres{j,1},caracter_r{i});
    end
    puntero=find(max(comp)==comp);
    patente=[patente ocr(puntero,1)];
end

set(handles.valor_1, 'String', patente(1,1:2));
set(handles.valor_2, 'String', patente(1,3:4));
```

```
set(handles.valor_3, 'String', patente(1,5:6));

% --- Executes on button press in cargar_imagen.
function cargar_imagen_Callback(hObject, eventdata, handles)
global im I
filename = uigetfile('*.jpg','Select an image file');
im = imread(filename);
% Scale data and display as image
imagesc(im);
axes(handles.axes2);
%% Convertir a escala de grises
I=rgb2gray(im);
%% Mostrar imagen
imshow(I);
pause (0.1);

%% Carga de caracteres numeros para el OCR
caracter0=(imread('caracteres\0.bmp'));
caracter1=(imread('caracteres\1.bmp'));
caracter2=(imread('caracteres\2.bmp'));
caracter3=(imread('caracteres\3.bmp'));
caracter4=(imread('caracteres\4.bmp'));
caracter5=(imread('caracteres\5.bmp'));
caracter6=(imread('caracteres\6.bmp'));
caracter7=(imread('caracteres\7.bmp'));
caracter8=(imread('caracteres\8.bmp'));
caracter9=(imread('caracteres\9.bmp'));

%% Carga de caracteres numeros para el OCR
caracterA=(imread('caracteres\A.bmp'));
caracterB=(imread('caracteres\B.bmp'));
caracterC=(imread('caracteres\C.bmp'));
caracterD=(imread('caracteres\D.bmp'));
caracterE=(imread('caracteres\E.bmp'));
caracterF=(imread('caracteres\F.bmp'));
caracterG=(imread('caracteres\G.bmp'));
caracterH=(imread('caracteres\H.bmp'));
caracterI=(imread('caracteres\I.bmp'));
caracterJ=(imread('caracteres\J.bmp'));
caracterK=(imread('caracteres\K.bmp'));
caracterL=(imread('caracteres\L.bmp'));
caracterM=(imread('caracteres\M.bmp'));
caracterN=(imread('caracteres\N.bmp'));
caracterO=(imread('caracteres\O.bmp'));
caracterP=(imread('caracteres\P.bmp'));
caracterQ=(imread('caracteres\Q.bmp'));
caracterR=(imread('caracteres\R.bmp'));
caracterS=(imread('caracteres\S.bmp'));
caracterT=(imread('caracteres\T.bmp'));
caracterU=(imread('caracteres\U.bmp'));
caracterV=(imread('caracteres\V.bmp'));
caracterW=(imread('caracteres\W.bmp'));
caracterX=(imread('caracteres\X.bmp'));
caracterY=(imread('caracteres\Y.bmp'));
caracterZ=(imread('caracteres\Z.bmp'));
```

```
 %% Redimensionado de caracteres número
caracteres{1,1}=imresize(caracter0,[68 40]);
caracteres{2,1}=imresize(caracter1,[68 40]);
caracteres{3,1}=imresize(caracter2,[68 40]);
caracteres{4,1}=imresize(caracter3,[68 40]);
caracteres{5,1}=imresize(caracter4,[68 40]);
caracteres{6,1}=imresize(caracter5,[68 40]);
caracteres{7,1}=imresize(caracter6,[68 40]);
caracteres{8,1}=imresize(caracter7,[68 40]);
caracteres{9,1}=imresize(caracter8,[68 40]);
caracteres{10,1}=imresize(caracter9,[68 40]);

%% Redimensionado de caracteres letras
caracteres{11,1}=imresize(caracterA,[68 40]);
caracteres{12,1}=imresize(caracterB,[68 40]);
caracteres{13,1}=imresize(caracterC,[68 40]);
caracteres{14,1}=imresize(caracterD,[68 40]);
caracteres{15,1}=imresize(caracterE,[68 40]);
caracteres{16,1}=imresize(caracterF,[68 40]);
caracteres{17,1}=imresize(caracterG,[68 40]);
caracteres{18,1}=imresize(caracterH,[68 40]);
caracteres{19,1}=imresize(caracterI,[68 40]);
caracteres{20,1}=imresize(caracterJ,[68 40]);
caracteres{21,1}=imresize(caracterK,[68 40]);
caracteres{22,1}=imresize(caracterL,[68 40]);
caracteres{23,1}=imresize(caracterM,[68 40]);
caracteres{24,1}=imresize(caracterN,[68 40]);
caracteres{25,1}=imresize(caracterO,[68 40]);
caracteres{26,1}=imresize(caracterP,[68 40]);
caracteres{27,1}=imresize(caracterQ,[68 40]);
caracteres{28,1}=imresize(caracterR,[68 40]);
caracteres{29,1}=imresize(caracterS,[68 40]);
caracteres{30,1}=imresize(caracterT,[68 40]);
caracteres{31,1}=imresize(caracterU,[68 40]);
caracteres{32,1}=imresize(caracterV,[68 40]);
caracteres{33,1}=imresize(caracterW,[68 40]);
caracteres{34,1}=imresize(caracterX,[68 40]);
caracteres{35,1}=imresize(caracterY,[68 40]);
caracteres{36,1}=imresize(caracterZ,[68 40]);

%% Asignacion de caracteres para ocr
ocr(1,1)='0';
ocr(2,1)='1';
ocr(3,1)='2';
ocr(4,1)='3';
ocr(5,1)='4';
ocr(6,1)='5';
ocr(7,1)='6';
ocr(8,1)='7';
ocr(9,1)='8';
ocr(10,1)='9';

ocr(11,1)='A';
ocr(12,1)='B';
```

```
ocr(13,1)='C';
ocr(14,1)='D';
ocr(15,1)='E';
ocr(16,1)='F';
ocr(17,1)='G';
ocr(18,1)='H';
ocr(19,1)='I';
ocr(20,1)='J';
ocr(21,1)='K';
ocr(22,1)='L';
ocr(23,1)='M';
ocr(24,1)='N';
ocr(25,1)='O';
ocr(26,1)='P';
ocr(27,1)='Q';
ocr(28,1)='R';
ocr(29,1)='S';
ocr(30,1)='T';
ocr(31,1)='U';
ocr(32,1)='V';
ocr(33,1)='W';
ocr(34,1)='X';
ocr(35,1)='Y';
ocr(36,1)='Z';

handles.caracteres=caracteres;
handles.ocr=ocr;

%% Fin Carga de caracteres

%% Normalizar Contraste
X=I;
[N,M] = size(X);
   Y = zeros(N,M);
   [i,j] = sort(X(:));
   z = zeros(N*M,1);
   d = fix(N*M/256+0.5);
   for i=1:255;
      z((i-1)*d+1:i*d) = (i-1)*ones(d,1);
   end
   z(255*d+1:N*M) = 255*ones(N*M-255*d,1);
   Y(j) = z;
im_g=X;

%% Binarización
umb=graythresh(im_g);
bw=imcomplement(im2bw(im_g,umb));
bw=medfilt2(bw, [8, 8], 'symmetric');
handles.bw=bw;

%% Etiquetar elementos conectados
[L Ne]=bwlabel(bw);
handles.Ne=Ne;

%% Calcular propiedades de los objetos de la imagen
```

```
propied= regionprops(L);
handles.propied=propied;
 propied.Area

for n=1:size(propied,1)
    if propied(n).Area > 1500 && propied(n).Area < 10000
    rectangle('Position',propied(n).BoundingBox,...
        'EdgeColor','g','LineWidth',2)
    pause (0.01)
    elseif propied(n).Area > 16000 && propied(n).Area < 110000
    rectangle('Position',propied(n).BoundingBox,...
        'EdgeColor','g','LineWidth',2)
    pause (0.01)

    end
end

pause (0.1)
hold off;
guidata(hObject,handles);

% --- Executes on button press in Patente_video.
function Patente_video_Callback(hObject, eventdata, handles)
%% Recorte patente de la imagen
bw=imcomplement(handles.bw);

Ic=bw;
st=strel('disk',10);                %Elemento estructurante de 10 pixeles de radio
IM2=imbothat(Ic,st);                  %Hace Bottom-Hat
I3=IM2;                            %Aplica Umbral
LH=strel('line',50,0);                %Elemento estructurante lineal horizontal
IM3=imclose(I3,LH);                     %Closing con elemento estructurante
LV=strel('line',20,140);              %Elemento estructurante lineal vertical
IM4=imopen(IM3,LV);                     %Hace opening con elemento estructurante
DIV=strel('line',70,90);              %Elemento estructurante lineal vertical
DIH=strel('line',40,0);               %Elemento estrucutrante lineal horizontal
IM5=imdilate(IM4,DIV);                  %Dilata con E.E. vertical
IM6=imdilate(IM5,DIH);                  %Dilata con E.E. horizontal
bw=IM6;

[L Ne]=bwlabel(bw);
propied=regionprops(L);
clear boundingbox;
for i=1:Ne

    boundingbox(i,:)=propied(i,:).BoundingBox;

end
proporcion=boundingbox(:,3)./boundingbox(:,4);
puntero=find(min(abs(proporcion-3.5))==abs(proporcion-3.5));
boxplaca=propied(puntero,:).BoundingBox;    %Tamaño inicial de la region

boxplaca=[...
    boxplaca(1,1)...
```

```
    boxplaca(1,2)...
    boxplaca(1,3)...
    boxplaca(1,4)];
handles.placa=imcrop(handles.bw,boxplaca);      %Realiza el corte
axes(handles.axes3);
imshow(handles.placa);                  %Muestra imagen de la zona de la placa
guidata(hObject,handles);

 %% Recortar Caracteres
placa=handles.placa;
caracteres=handles.caracteres;
ocr=handles.ocr;
[L Ne]=bwlabel(placa);
handles.Ne=Ne;
disp(Ne);

%% Calcular propiedades de los objetos de la imagen
propied= regionprops(L);
handles.propied=propied;
hold on;

%% Graficar las 'cajas' de frontera de los objetos
for n=1:size(propied,1)
     if propied(n).Area > 500 && propied(n).Area < 2000
    rectangle('Position',propied(n).BoundingBox,...
        'EdgeColor','g','LineWidth',2)
    pause (0.01)
    end
end
pause (0.1)
hold off;
guidata(hObject,handles);
clear boundingbox;
for i=1:Ne
    boundingbox(i,:)=propied(i,:).BoundingBox;
end
proporcion=boundingbox(:,3)./boundingbox(:,4);
disp(proporcion);
puntero=find(proporcion>0.82 | proporcion<0.2);
proporcion(puntero)=0;
puntero=find(proporcion~=0);
[f c]=size(puntero);
disp(puntero);
caracter={};
for i=1:f
    boxplaca=propied(puntero(i,1),:).BoundingBox;
    caracter{i}=imcrop(placa,boxplaca);
end
disp(caracter);
[f c]=size(caracter);
sizes=[];
for i=1:c
    sizes=[sizes; size(caracter{i})];
end
disp(sizes);
```

```
[orden indice]=sort(sizes,1,'descend')
indice2=indice(1:6,1)
disp(indice2);
indice2=sort(indice2);
disp(indice2(:,1));
caracte2={};
for i=1:6
    caracter2{i}=caracter{indice2(i,1)};
end
disp(caracter2);

%% Escalar caracteres al tamaño del patrón o template
for i=1:6
    caracter_r{i}=imresize(caracter2{i},[68 40]);
end

%% Reconocimiento de caracteres
[f c]=size(caracter_r);
comp=zeros(15,1);
puntero=0;
patente=[''];

comp=0;
for i=1:2
    for j=11:36
        comp(j,1)=corr2(caracteres{j,1},caracter_r{i});
    end
    puntero=find(max(comp)==comp);
    patente=[patente ocr(puntero,1)];
end

comp=0;
for i=3:4
    for j=1:36
        comp(j,1)=corr2(caracteres{j,1},caracter_r{i});
    end
    puntero=find(max(comp)==comp);
    patente=[patente ocr(puntero,1)];
end

comp=0;
for i=5:6
    for j=1:10
        comp(j,1)=corr2(caracteres{j,1},caracter_r{i});
    end
    puntero=find(max(comp)==comp);
    patente=[patente ocr(puntero,1)];
end

set(handles.valor_1, 'String', patente(1,1:2));
set(handles.valor_2, 'String', patente(1,3:4));
set(handles.valor_3, 'String', patente(1,5:6));
```