
Projecte VC: Detecció de Matrícules.

Table of Contents

Imatges Inicials	1
Detectem Potencials Matrícules	1
Eliminem no-matrícules	3
Seleccionem Números	4

Imatges Inicials

```
close all; clc
w = warning ('off','all');

f=dir(['cars' '/*.jpg']);
files={f.name};
numcotxes = numel(files);
names = convertCharsToStrings(files);
for k=1:numcotxes
    names(k) = erase(names(k), ".jpg");
end
im_ors=cell(1,numcotxes);
for k=1:numcotxes
    im_ors{k}=imread(files{k});
end

close all;
```

Detectem Potencials Matrícules

```
% Aquesta funció detecta aquelles regions de la imatge que són potencials
% matrícules

function immat = F_PotencialsMatricules(imor)

    % Passem a Blanc i negre i obtenim edges
    imgrey = rgb2gray(imor);
    imbin = imbinarize(imgrey);
    imedges = edge(imgrey, 'prewitt');
    imedges(200,:) = 0;

    % Seleccionem les zones tancades
    ee = strel('rectangle',[5,10]);
    immat = imdilate(imedges,ee);
    immat = imfill(immat,"holes");
    ee = strel('line',20,0);
    immat = imerode(immat,ee);
```

```

ee = strel('line',10,90);
immat = imeroode(immat,ee);

% Eliminem els bordes de les zones massa grans
ee = strel('rectangle',[200,200]);
imgrans = imopen(immat,ee);
imgrans = imreconstruct(imgrans,immat);
imgransbordes = imdilate(imgrans,strel('disk',10));
imgransbordes = imgransbordes - imgrans;
imedges = logical(imedges - (imedges .* imgransbordes));

% Repetim selecció de zones tancades
ee = strel('rectangle',[5,10]);
immat = imdilate(imedges,ee);
immat = imfill(immat,"holes");
ee = strel('line',20,0);
immat = imeroode(immat,ee);
ee = strel('line',10,90);
immat = imeroode(immat,ee);

% Eliminem les línies verticals o horitzontals molt finetes
immat = imopen(immat,strel("rectangle",[15,1]));
immat = imopen(immat,strel("rectangle",[1,15]));

% Separem blobs propers
td = bwdist(~immat);
td = imhmax(td,4);
segm = watershed(-td);
immat = ~(~immat | (segm == 0));

% Eliminem les zones no rectangulars
ee = strel('rectangle',[15,50]);
imnorrectangles = imopen(immat,ee);
imnorrectangles = imreconstruct(imnorrectangles,immat);

immat = logical(immat .* imnorrectangles);

% Eliminem les zones sense línies verticals o horitzontals
ee = strel('rectangle',[5,1]);
imlines_v = imopen(imedges,ee);
imlines = imreconstruct(imlines_v,immat);
ee = strel('rectangle',[1,5]);
imlines_h = imopen(imedges,ee);
imlines = imreconstruct(imlines_h,imlines);

immat = logical(immat .* imlines);

% Eliminem els blobs en contacte amb els bordes
[n, m] = size(imbin);
imbordes = logical(zeros(n,m));
imbordes(1:3,:) = 1;
imbordes(n-2:n,:) = 1;
imbordes(:,1:3) = 1;
imbordes(:,m-2:m) = 1;

```

```

imtouching = imreconstruct(imbordes,immat);

immat = logical(immat - imtouching);
end

im_mat=cell(1,numcotxes);
for k=1:numcotxes
    im_mat{k}=F_PotencialsMatricules(im_ors{k});
end

% Exemples:
for k=1:3
    imres = im_ors{k};
    immat2 = imdilate(im_mat{k},strel("disk",5)) - im_mat{k};
    imres(:,:,:3) = imres(:,:,:3) .* uint8(~immat2);
    imres(:,:,:1) = imres(:,:,:1) .* uint8(~immat2);
    imres(:,:,:2) = imres(:,:,:2) + uint8(immat2)*256;
    figure, imshow(imres);
end

close all;

```

Eliminem no-matricules

```

% Eliminem moltes matricules fals positives que hem detectat
% en el pas anterior

function immat = F_EliminarNoMatricules(imor,immator)
[n, m] = size(immator);
immat = logical(zeros(n,m));

Iprops=regionprops(immator,'BoundingBox','Area','Image');
count = numel(Iprops);

for i=1:count
    boundingBox=Iprops(i).BoundingBox;
    region = imcrop(rgb2gray(imor),boundingBox);

    ridncalv = @ridncalv;
    regbin = ~imbinarize(region,ridncalv(region));

    regionmat = imcrop(immator,boundingBox);
    regbin = regionmat.*regbin;

    % Eliminem les regions amb < 5 regions connexes,
    % després de filtrar estructures petites
    ee = strel("rectangle",[7,1]);
    regero = imerode(regbin,ee);
    regrec2 = imreconstruct(regero,regbin);

    ee = strel("rectangle",[1,2]);

```

```

regero = imerode(regrec2,ee);
regrec2 = imreconstruct(regero,regrec2);

regrec2 = imclose(regrec2,strel("rectangle",[2,1]));
regrec2 = imclose(regrec2,strel("rectangle",[1,2]));

regrec2 = padarray(regrec2,[1,1],1);
cc = bwconncomp(regrec2);

% Eliminem les regions amb proporcions inadequades
[n2, m2] = size(regbin);
ratio = abs(n2/m2 - 110/520)*100;

%figure, imshow(regrec2),title(["cc", cc.NumObjects, "ratio", ratio]);

% Afegim la potencial matricula
if cc.NumObjects > 4 && ratio < 35
    X = boundingBox(1); Y = boundingBox(2);
    W = boundingBox(3); H = boundingBox(4);
    immat(Y:Y+H, X:X+W) = regionmat;
end

end
end

im_mat2=cell(1,numcotxes);
for k=1:numcotxes
    im_mat2{k}=F_EliminarNoMatricules(im_ors{k},im_mat{k});
end

% Exemples:
for k=1:3
    imres = im_ors{k};

    immat2 = imdilate(im_mat{k},strel("disk",5)) - im_mat{k};
    imres(:,:,:,3) = imres(:,:,:,:3) .* uint8(~immat2);
    imres(:,:,:,:2) = imres(:,:,:,:2) .* uint8(~immat2);
    imres(:,:,:,:1) = imres(:,:,:,:1) + uint8(immat2)*256;

    immat2 = imdilate(im_mat2{k},strel("disk",5)) - im_mat2{k};
    imres(:,:,:,:3) = imres(:,:,:,:3) .* uint8(~immat2);
    imres(:,:,:,:1) = imres(:,:,:,:1) .* uint8(~immat2);
    imres(:,:,:,:2) = imres(:,:,:,:2) + uint8(immat2)*256;

    figure, imshow(imres);
end

close all;

```

Seleccionem Números

```
% Seleccionem els números dels components connexos de les matrícules

function imnums = F_SeleccionarNumeros(imor,immat)
[n, m] = size(immat);
imnums = logical(zeros(n,m));

Iprops=regionprops(immat,'BoundingBox','Area','Image');
count = numel(Iprops);

for i=1:count
    if Iprops(i).Area < 1000
        continue
    end
    boundingBox=Iprops(i).BoundingBox;
    regionCol = imcrop(imor,boundingBox);
    region = rgb2gray(regionCol);

    ridncalv = @ridncalv;
    regbin = ~imbinarize(region,ridncalv(region));

    regionmat = imcrop(immat,boundingBox);
    regbin = regionmat.*regbin;

    % Eliminem els elements molt grans
    reggran = imopen(regbin,strel("rectangle",[9,9]));
    reglletres = logical(regbin - reggran);

    % Eliminem els elements molt llargs
    reggran = imopen(reglletres,strel("rectangle",[1,30]));
    reglletres = logical(reglletres - reggran);

    %Eliminem les zones de colors(marcadors de pais)
    imhsv = rgb2HSV(regionCol);
    imcol = imhsv(:,:,2) .* imhsv(:,:,3);
    imcol = imbinarize(imcol,0.3);
    reglletres = reglletres .* (~imcol);
    reglletres = logical(reglletres);

    %Filtrem els components connexos molt estrets
    imer = imeroide(reglletres, strel("rectangle",[5,1]));
    reglletres = imreconstruct(imer,reglletres);

    % Unim lletres separades
    reglletres = imclose(reglletres,strel("rectangle",[4,1]));

    % Eliminem els components connexos molt petits
    reglletres = bwareafilt(reglletres,[15,100000]);

    % Treballem amb les regions
    [n2,m2] = size(reglletres);
    reglletres2 = reglletres;

props=regionprops(bwconncomp(reglletres),'BoundingBox','PixelList','Solidity','Circularit
    for j=1:numel(props)
```

```

prop = props(j);
pl = prop.PixelList;
X = ceil(prop.BoundingBox(1)); Y = ceil(prop.BoundingBox(2));
W = prop.BoundingBox(3); H = prop.BoundingBox(4);

% Eliminem elements poc sòlids
if prop.Solidity < 0.3
    reglletres = F_PintaPixels(reglletres,pl,0);
    continue;
end

% Eliminem els elements massa llargs
if W > m2*0.4
    reglletres = F_PintaPixels(reglletres,pl,0);
    continue;
end

% Eliminem els elements massa circulars
if prop.Circularity > 0.5 && W/H > 0.9
    reglletres = F_PintaPixels(reglletres,pl,0);
    continue;
end

% Dividim els elements que no encaixen la proporció
ratio = 0.9;
if W/H > ratio && numel(props) < 8
    reglletres(Y:Y+H,X+ceil(W/2)) = 0;
end

end

X = boundingBox(1); Y = boundingBox(2);
W = boundingBox(3); H = boundingBox(4);
imnums(Y:Y+H, X:X+W) = reglletres;
end
end

im_nums=cell(1,numcortexes);
for k=1:numcortexes
    im_nums{k}=F_SeleccionarNumeros(im_ors{k},im_mat2{k});
end

% Exemples:
for k=1:numcortexes
    imres = im_ors{k};
    immat2 = imdilate(im_nums{k},strel("disk",5)) - im_nums{k};
    imres(:,:,:2) = imres(:,:,:2) .* uint8(~immat2);
    imres(:,:,:1) = imres(:,:,:1) .* uint8(~immat2);
    imres(:,:,:3) = imres(:,:,:3) + uint8(immat2)*256;
    figure, imshow(imres);
end

```







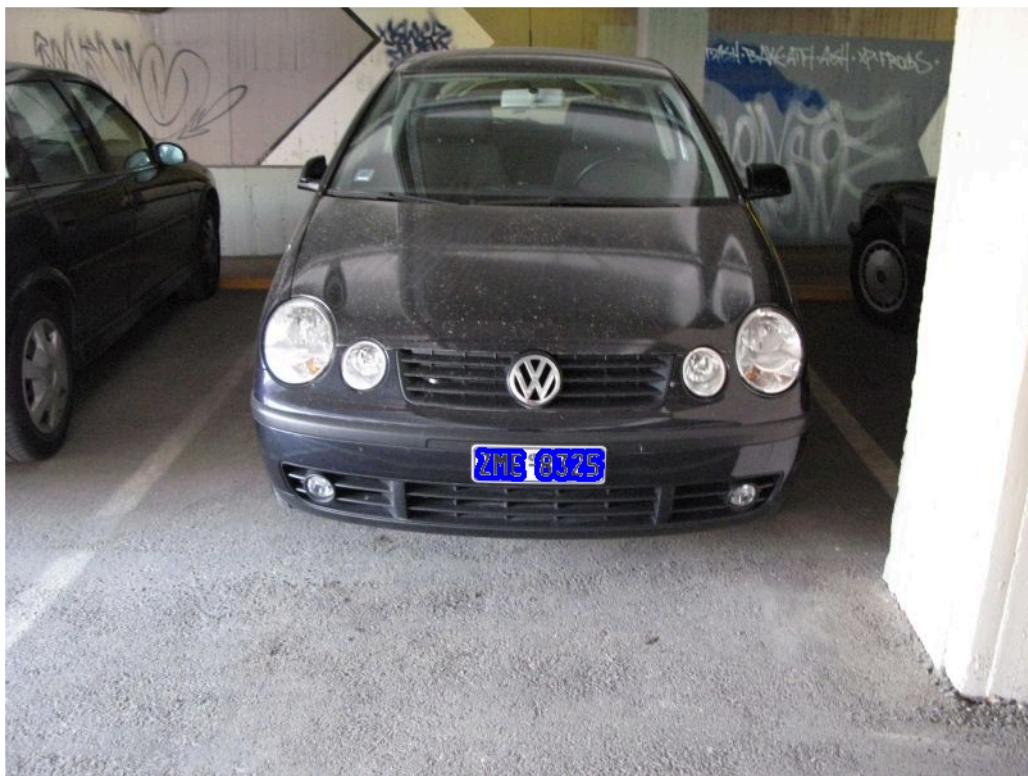




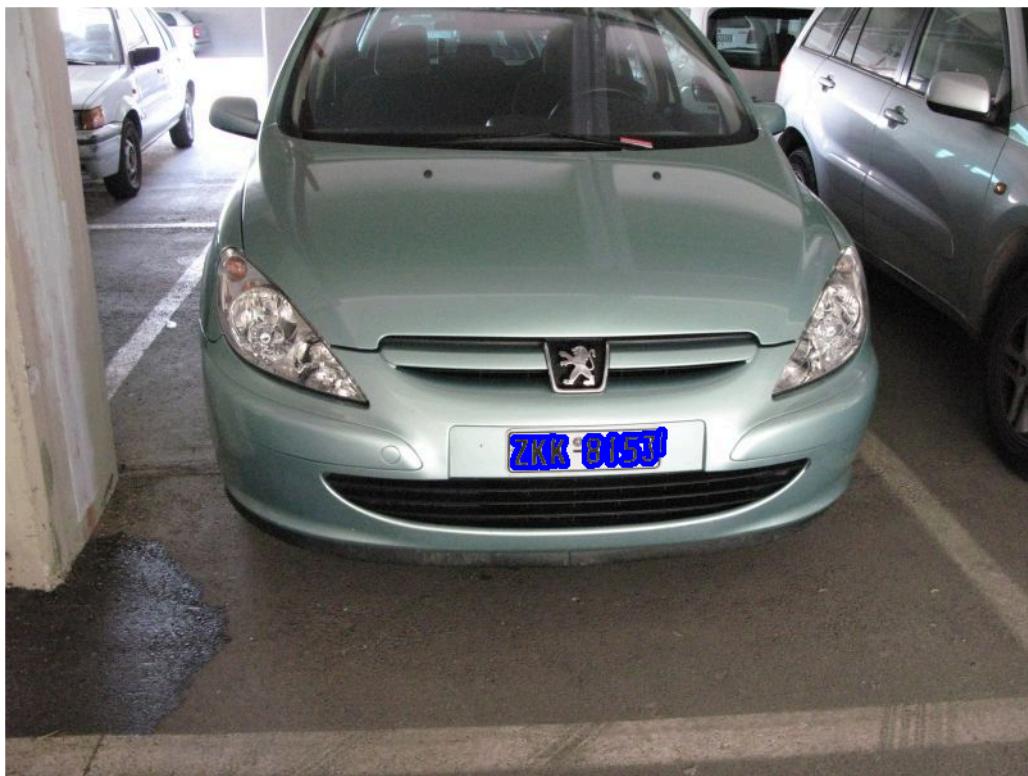




































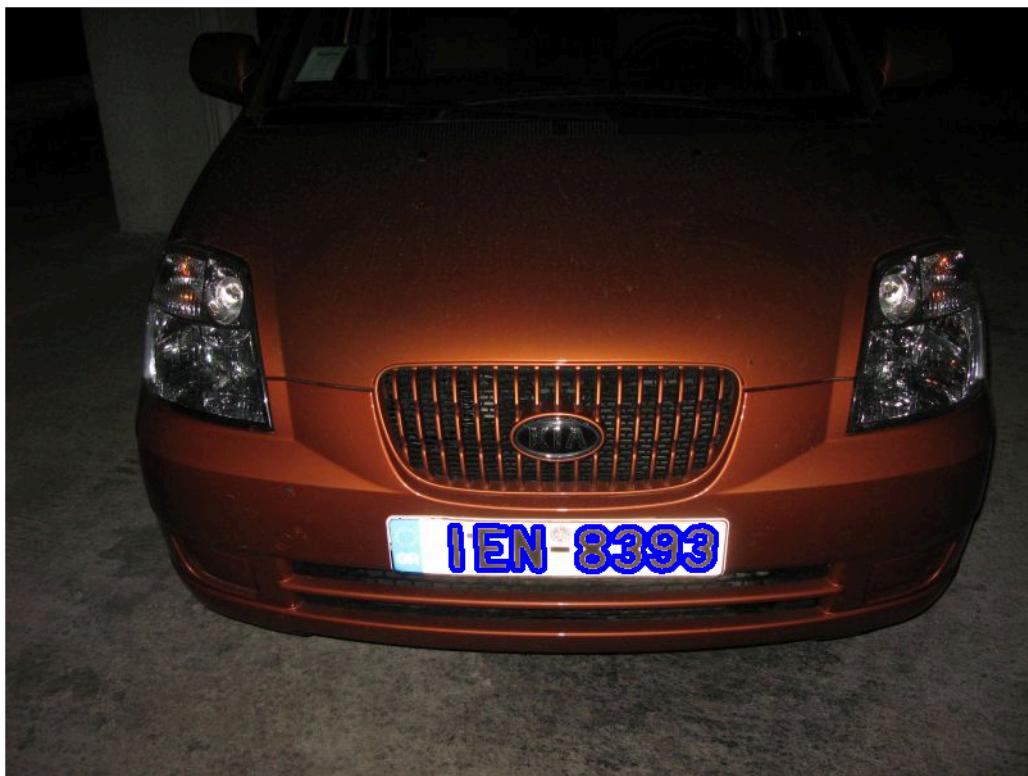
























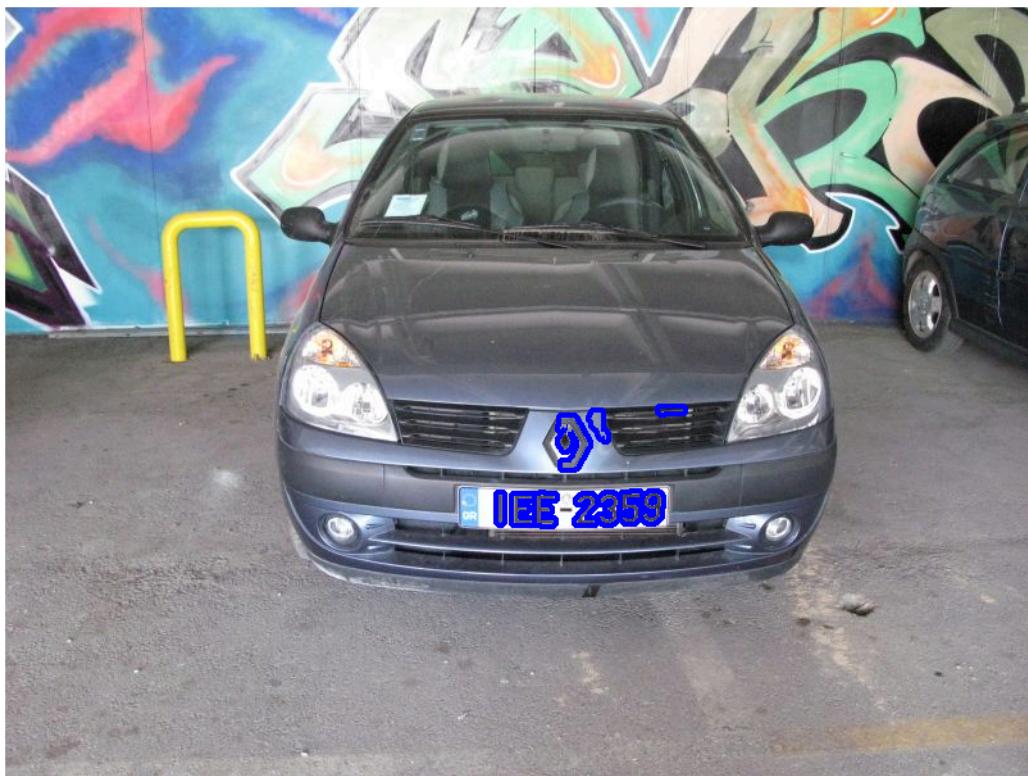








































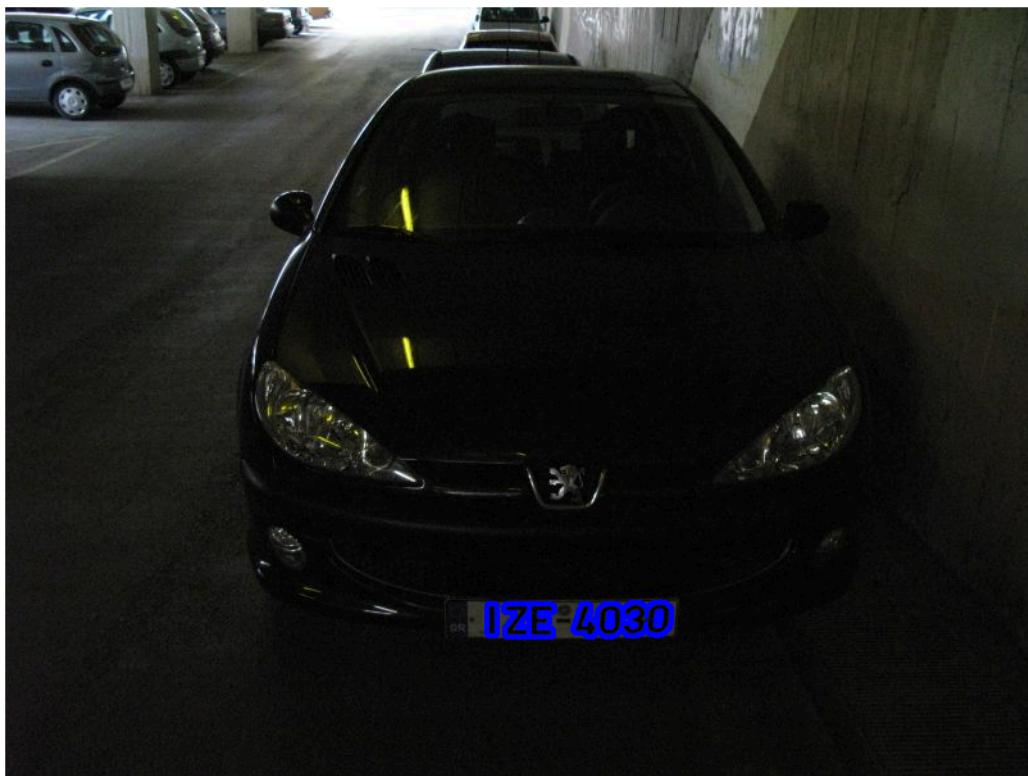
























Published with MATLAB® R2022b