Script for processing the photos taken of the product number 2

Pre-processing

```
% Include actual folder in the matlab path
addpath(pwd);

% Load pre-trained Convolutional Neural Network
load('trained_CNN.mat');

% Navigate to the photos
cd fotos21dec17/
cd produs2/
cd original

% Choose to plot the results
PLOT = 0;
```

Read and iterate through the photos

```
files = dir('*.jpg');
for num_file = 1:length(files)

    % Read the file
    I = imread(files(num_file).name);
    if PLOT
        figure, imshow(I)
    end
    % to gray
    Ig = rgb2gray(I);
    % to black and white
    Ibw = im2bw(Ig,graythresh(Ig));

    % erode holes
    Ibw2 = bwareaopen(Ibw, 80, 8);
    SE = strel('disk',4);
    Ibw2 = imerode(Ibw2,SE);

    % Find conected region and their properties
    stats = regionprops(Ibw2,'boundingbox','Area','Perimeter');
    % order by area
    Afields = fieldnames(stats);
    Acell = struct2cell(stats);
    sz = size(Acell);
    % Convert to a matrix
    Acell = reshape(Acell, sz(1), []); % Px(MxN)
    % Make each field a column
    Acell = Acell';                    % (MxN)xP
    % Sort by first field "AREA"
```

```
Acell = sortrows(Acell, 1, 'descend');
% Put back into original cell array format
Acell = reshape(Acell', sz);
% Convert to Struct
Asorted = cell2struct(Acell, Afields, 1);

if PLOT
    % Draw the largest square
    figure,imshow(Ibw); hold on;
    rectangle('position',Asorted(1).BoundingBox,'edgecolor','r','linewidth',2);
    hold off
end
```

Choose between circular and square plates

Compute circularity

```
circularity = Asorted(1).Perimeter^2 / (4 * pi * [Asorted(1).Area]);
```

CIRCULAR case

```
if circularity  < 1.15
    disp(files(num_file).name)

    % Cut circular plates (1689*1686)
    I1 = imcrop(I,Asorted(1).BoundingBox);

    % fix size Bouondng Box [Xmin Ymin Width Weight]
    BB = [330 330 1020 1020];
    Ic = imcrop(I1, BB);
    if PLOT
        figure, imshow(I1)
    end

    % to black and white
    Ibw = im2bw(rgb2gray(Ic),graythresh(rgb2gray(Ic)));

    % find inner circle
    stats = regionprops('table',Ibw,'Centroid','MajorAxisLength','MinorAxisLength
    diameters = mean([stats.MajorAxisLength stats.MinorAxisLength],2);
    big_diam = diameters(diameters > 950,:);
    centers = stats.Centroid(diameters > 950);
    center_image = size(Ibw)/2;

    % compute the distance to the centre
    dist_center=[];
    for i=1:1:size(centers,1)
      dist_center(i) = sqrt(sum(center_image - centers(i,:)).^2);
    end
    [min_dist, ind] = min(dist_center);
```

```
if PLOT
    figure, imshow(Ibw), hold on
    viscircles(center_image,big_diam(ind)/2) , hold off
end

% Choose found circles by radius
radii = big_diam(ind)/2;
rad_offset = 25;
if radii > 500
    rad_offset = 80;
end
for i=1:1:size(Ic,1)
    for j=1:1:size(Ic,2)
        if sqrt(sum((center_image - [i,j]).^2)) >= (radii-rad_offset)
            Ic(i,j,:) = 255;
        end
    end
end

% Find Letters Area
Ig = rgb2gray(Ic);
Ig_ad = imadjust(Ig);
% Edge detection
I_edge = edge(Ig_ad,'Canny',0.16827,2.0);
% Eliminate small objects
I_edge = bwareaopen(I_edge, 60, 8);
% Dilate edge
SE  = strel('Disk',10,4);
I_edge = imdilate(I_edge, SE);

% Find conected region and their properties
stats = regionprops(I_edge, 'Area','BoundingBox','Orientation');
Afields = fieldnames(stats);
Acell = struct2cell(stats);
minX=size(Ic,1);
minY=size(Ic,2);
maxX=0;
maxY=0;
for ii=size(stats,1):-1:1
    if (stats(ii).Area > 18000)
        Acell(:,ii)=[];
    else
        minX=min(stats(ii).BoundingBox(1),minX);
        minY=min(stats(ii).BoundingBox(2),minY);
        maxX=max(stats(ii).BoundingBox(1)+stats(ii).BoundingBox(3),maxX);
        maxY=max(stats(ii).BoundingBox(2)+stats(ii).BoundingBox(4),maxY);
    end
end
AA = cell2struct(Acell, Afields, 1);

% Cut around letters [Xmin Ymin Width Weight]
I_letters=imcrop(Ic, [minX minY (maxX-minX) (maxY-minY)]);
if PLOT
```

```
        figure, imshow(I_letters)
        figure(20),clf, imshow(I_letters)
        figure(10),clf, subplot(2,1,1), imshow(Ic), ...
            subplot(2,1,2), imshow(I_edge), hold on, ...
            for i = 1:size(AA,1)
                rectangle('Position', AA(i).BoundingBox,'EdgeColor','r', 'linewidt
            end
        hold off
    end

    % to grayscale
    Ig = rgb2gray(I_letters);
    boxes=[];
    disk_size=2;
    canny_thesh = 0.475;
    tries = 1;
    white_points = [1400 1500 1600];
    % We look for 3 groups of letters
    while(size(boxes,1) ~= 3) || (sum(sum(aux)) < 50)
        if(disk_size > 8)
            disk_size=2;
        end
        % Compute edges
        I_edge = edge(Ig,'Canny',canny_thesh,1.95);
        while (sum(sum(I_edge)) < white_points(tries))
            canny_thesh = canny_thesh-0.025;
            I_edge = edge(Ig,'Canny',canny_thesh,1.95);
        end

        % Open edges
        disk_size = disk_size + 1;
        SE  = strel('Disk',disk_size,8);
        I_edge = imdilate(I_edge, SE);
        % Add a black frame
        aux = zeros(size(I_edge) + [2 2]);
        aux(2:size(I_edge,1)+1,2:size(I_edge,2)+1) = I_edge;
        I_edge = aux;

        % compute image labels, using minimal connectivity
        lbl = bwlabel(I_edge, 4);
        nLabels = max(lbl(:));

        % Compute enclosing oriented boxes
        boxes = imOrientedBox(lbl);
        % Filter the boxes by the expected area
        for i=size(boxes,1):-1:1
            if (boxes(i,3)*boxes(i,4)<1100) || ...
                (boxes(i,3)*boxes(i,4)>7000)
                boxes(i,:)=[];
            end
        end
        if PLOT && nLabels > 0
            rgb = label2rgb(lbl, jet(nLabels), 'w', 'shuffle');
```

```
            figure, imshow(I_edge), hold on;
            drawOrientedBox(boxes, 'linewidth', 2);
        end
    end

    % Add area value
    boxes = [boxes  zeros(size(boxes,1),1)];
    for p=1:1:size(boxes,1)
        boxes(p,size(boxes,2)) = boxes(p,3)*boxes(p,4);
    end
    % Find order by area
    index = [0 0 3 2 1];
    [min_a,small]=min(boxes(:,6));
    [max_a,large]=max(boxes(:,6));
    med = index(small+large);

    cx    = boxes(:,1);
    cy    = boxes(:,2);
    hl    = boxes(:,3) /2;
    hw    = boxes(:,4) /2;
    theta = boxes(:,5);

    keys={};
    df = 26;  % letter height

    n=1;
    letters = {};
    groups=[];
    for i=1:size(boxes,1)
        % pre-compute angle data
        cot = cosd(theta(i));
        sit = sind(theta(i));
        % x and y shifts
        lc = hl(i) * cot;
        ls = hl(i) * sit;
        wc = hw(i) * cot;
        ws = hw(i) * sit;
        % coordinates of box vertices
        vx = cx(i) + [-lc + ws; lc + ws ; lc - ws ; -lc - ws];
        vy = cy(i) + [-ls - wc; ls - wc ; ls + wc ; -ls + wc];
        tam = [max(vx)-min(vx), max(vy)-min(vy)];
        if max(tam) < 56
            tam(find(tam==max(tam))) = 56;
        end
        if min(tam) < 53
            tam(find(tam==min(tam))) = 53;
        end
        Ic=imcrop(I_letters,[max(min(vx),1), max(min(vy),1), tam(1), tam(2)]);
        Ir=imrotate(Ic, theta(i));

        % Fix size rectangles
        if small == i
            disp('rotar')
```

```
    Ir=imrotate(Ir, 90);
    dc = 15;
elseif large == i && (boxes(large,6)/boxes(med,6) > 1.15)
        dc = 52;
else
        dc = 40;
end
vf = floor(size(Ir,1)/2) + [df -df];
vc = floor(size(Ir,2)/2) + [dc -dc];
keys{i}=imcrop(Ir,[max(min(vc),1), max(min(vf),1), max(vc)-min(vc), max(vf

% Segment indivisual letters studying the projection of borders into X-axi
img=keys{i};
Ig = rgb2gray(img);
J1 = histeq(Ig);
K1 = wiener2(J1,[3 3]);
J2 = histeq(K1);
K2 = wiener2(J2,[3 3]);
% values addjustest with seg. app
imgout = imadjust(K2,[0.03; 0.92],[0.00; 1.00],2.88);
thresholds = multithresh(imgout,8);
[~,quantIndex] = imquantize(imgout,thresholds);
mask = ismember(quantIndex,[9]);
Ibw = bwareaopen(mask, 4,4);
SE  = strel('Disk',1,4);
I_edge = imdilate(Ibw, SE);

% Projection
YProj = sum(I_edge,1);
if PLOT
    figure,plot(YProj)
end
% Look for data falling to zero
ind = find([0,diff((YProj == 0))>0] & (YProj == 0));
% eliminate last pixels valley
if (sum(YProj(ind(size(ind,2)):size(YProj,2))) == 0) || (ind(size(ind,2))
    ind(size(ind,2))=[];
end

% Use length as indicator of the number of letters
if size(keys{i},2) == 105 % 4 letters
    groups(i) = 4;
    % If case there are more valleys than necessary
    while size(ind,2) > 3
        % eliminate the smallest valley
        sum_ind = [];
        for valley = 1:1:size(ind,2)
            p=ind(valley);
            while YProj(p) == 0
                p=p+1;
            end
            sum_ind(valley) = p - ind(valley);
        end
```

```
                if sum(sum_ind==min(sum_ind)) == 1
                    [a b]=min(sum_ind);
                else
                    edge_ind=[];
                    for t=1:size(sum_ind,2)
                        edge_ind(t) = min(abs(ind(t)-1),abs(ind(t)-size(YProj,2)))
                    end
                    edge_ind(find(edge_ind==0)) = size(YProj,2);
                    [a b]=min(edge_ind);
                end
                ind(b) = [];
            end
            % First letter
            col = 1;
            while YProj(col) == 0
                col= col+1;
            end
            letters{n} = img(:,max(col-1,1):min(max(col-1,1)+25,size(YProj,2)),:);
            if PLOT
                figure,imshow(letters{n})
            end
            n=n+1;
            % next letters
            for j=1:3
                col = ind(j);
                while YProj(col) == 0
                    col= col+1;
                end
                letters{n} = img(:,max(col-1,1):min(max(col-1,1)+25,size(YProj,2)),:);
                if PLOT
                    figure,imshow(letters{n})
                end
                n=n+1;
            end
        % 3 letters
        elseif size(keys{i},2) > 70 %== 81
            groups(i) = 3;
            % If there are more valleys than necessary
            while size(ind,2) > 2
               % eliminate the smallest valley
                sum_ind = [];
                for valley = 1:1:size(ind,2)
                    p=ind(valley);
                    while YProj(p) == 0
                        p=p+1;
                    end
                    sum_ind(valley) = p - ind(valley);
                end
                if sum(sum_ind==min(sum_ind)) == 1
                    [a b]=min(sum_ind);
                else
                    edge_ind=[];
                    for t=1:size(sum_ind,2)
```

```
                        edge_ind(t) = min(abs(ind(t)-1),abs(ind(t)-size(YProj,2)))
                    end
                    edge_ind(find(edge_ind==0)) = size(YProj,2);
                    [a b]=min(edge_ind);
                end
                ind(b) = [];
            end
            % First letter
            col = 1;
            while YProj(col) == 0
                col= col+1;
            end
            letters{n} = img(:,max(col-1,1):min(max(col-1,1)+25,size(YProj,2)),:);
            if PLOT
                figure,imshow(letters{n})
            end
            n=n+1;
            % next letters
            for j=1:2
                col = ind(j);
                while YProj(col) == 0
                    col= col+1;
                end
                letters{n} = img(:,max(col-1,1):min(max(col-1,1)+25,size(YProj,2)),:);
                if PLOT
                    figure,imshow(letters{n})
                end

                n=n+1;
            end
        % 1 letter
        else
            groups(i) = 1;
            col = 1;
            while YProj(col) == 0
                col = col+1;
            end
            letters{n} = img(:,max(col-1,1):min(max(col-1,1)+25,size(YProj,2)),:);
            if PLOT
                figure,imshow(letters{n})
            end
            n=n+1;
        end
    end
```

RECTANGULAR case

```
elseif circularity < 1.8
    message = sprintf('Im: %f, Circularity: %.3f, so the object is a rectangle', n
    disp(message);
    % Orientate the image
    Ibw = imcrop(Ibw,Asorted(1).BoundingBox);
```

```
        Ibw = imfill(Ibw,'holes');
        [Gmag, Gdir] = imgradient(Ibw,'sobel');
        [Gx, Gy] = imgradientxy(Ibw,'sobel');
        direction_mat = Gy(1:50,400:1400);

        % direccion de la linea
        [H,theta,rho] = hough(direction_mat,'RhoResolution',1,'ThetaResolution',0.05);
        peaks  = houghpeaks(H,5);
        lines = houghlines(direction_mat,theta,rho,peaks);
        angle = sum(lines(:).theta/numel(lines))-sign(sum(lines(:).theta/numel(lines))
        %  Recortar (922*1840) y Rotar
        I_crop = imcrop(I,Asorted(1).BoundingBox);
        I_rot = imrotate(I_crop,angle);

        % Cut fix size [Xmin Ymin Width Weight]
        BB = [155 180 490 610; 670 180 490 610; 1210 180 490 610];
        Ic1 = imcrop(I_rot, BB(1,:));
        Ic2 = imcrop(I_rot, BB(2,:));
        Ic3 = imcrop(I_rot, BB(3,:));
        figure,imshow(I_rot), hold on;
        rectangle('position', BB(1,:),'edgecolor','r','linewidth',2);
        rectangle('position', BB(2,:),'edgecolor','b','linewidth',2);
        rectangle('position', BB(3,:),'edgecolor','g','linewidth',2);
        hold off;
        figure,subplot(1,3,1),imshow(Ic1),subplot(1,3,2),imshow(Ic2),subplot(1,3,3),im

    else
        message = sprintf('Wrong detection');
    end

    labels=[];
    scores=[];
    for m = 1:1:size(letters,2)
        grayIm=letters{m};
        [label, score] = classify(convnet,grayIm(:,:,1));
        labels = [labels label];
        scores = [scores max(score)];
    end
    % g1
    count=0;
    g=char(labels(1:groups(1)));
    for p=1:1:size(g,1)
        if ( (strcmp('1r',g(p,:))) || (strcmp('2r',g(p,:))) || ...
             (strcmp('3r',g(p,:))) || (strcmp('4r',g(p,:))) || ...
             (strcmp('5r',g(p,:))) || (strcmp('6r',g(p,:))) || ...
             (strcmp('7r',g(p,:))) || (strcmp('8r',g(p,:))) || ...
             (strcmp('9r',g(p,:))) )
            count=count+1;
        end
        g1(p,1)= g(p,1);
    end
    if count > (size(g1,2)/2)
       g1=fliplr(g1);
```

```
    end
    % g2
    count=0;
    g=char(labels(1+groups(1):groups(1)+groups(2)));
    for p=1:1:size(g,1)
        if ( (strcmp('1r',g(p,:))) || (strcmp('2r',g(p,:))) || ...
             (strcmp('3r',g(p,:))) || (strcmp('4r',g(p,:))) || ...
             (strcmp('5r',g(p,:))) || (strcmp('6r',g(p,:))) || ...
             (strcmp('7r',g(p,:))) || (strcmp('8r',g(p,:))) || ...
             (strcmp('9r',g(p,:))) )
            count=count+1;
        end
        g2(p,1)= g(p,1);
    end
    if count > (size(g2,2)/2)
        g2=fliplr(g2);
    end
    % g3
    count=0;
    g=char(labels(1+groups(1)+groups(2):groups(1)+groups(2)+groups(3)));
    for p=1:1:size(g,1)
        if ( (strcmp('1r',g(p,:))) || (strcmp('2r',g(p,:))) || ...
             (strcmp('3r',g(p,:))) || (strcmp('4r',g(p,:))) || ...
             (strcmp('5r',g(p,:))) || (strcmp('6r',g(p,:))) || ...
             (strcmp('7r',g(p,:))) || (strcmp('8r',g(p,:))) || ...
             (strcmp('9r',g(p,:))) )
            count=count+1;
        end
        g3(p,1)= g(p,1);
    end
    if count > (size(g3,2)/2)
        g3=fliplr(g3);
    end

    message = sprintf('Letters from file %d', num_file);
    disp(message)
    g1'
    g2'
    g3'
end

 cd ..
 cd ..
 cd ..
```