Assignment 5: Normalized Cross Correlation

```
close all; clc; clear all;

% Gathers BW image from location . . .
BW = threshold_image('xcorr/unknown.jpg');

% Provides a set of objects from BW image . . .
[~, UnknownImagesScaled] = process_objects(BW);

% [ Repeat with template ]
BW = threshold_image('xcorr/template.jpg');
[~, TemplateImagesScaled] = process_objects(BW);

% Create empty arrays for storing max correlation and index . . .
maxCorr = [];
maxIndex = [];

% Goes through all images and templates to compare . . .
for i = UnknownImagesScaled
    corr = [];
    for j = TemplateImagesScaled
        % Finds how much the object correlates with the
 templates . . .
        value = normxcorr2(i{1}, j{1});
        corr(end + 1) = max(value(:));
    end

    % Stores maximum correlation and index . . .
    [maxCorr(end+1), maxIndex(end+1)] = max( corr(:) );
end


%% Function: process_objects()

function [subImages, subImagesScaled] = process_objects(BW)

    % Gathering objects using regionprops . . .
    [labels, ~] = bwlabel(BW, 8);
    Istats = regionprops(labels, 'basic', 'Centroid');

    % Cutting out smaller objects . . .
    Istats( [Istats.Area] < 1000 ) = [];
    num = length( Istats );

    % Create bounding box around objects . . .
    Ibox = floor( [Istats.BoundingBox] );
    Ibox = reshape( Ibox, [4 num]);

    % Plot bounding boxes . . .
    for k = 1 : num
        col_1 = Ibox(1, k);
```

```
        col_2 = col_1 + Ibox(3, k);
        row_1 = Ibox(2, k);
        row_2 = row_1 + Ibox(4, k);

        subImages{k} = BW( row_1:row_2, col_1:col_2 );
        subImagesScaled{k} = imresize(subImages{k}, [24 12]);
    end
end


%% Function: threshold_image()

function BW = threshold_image(name)

    % Open image and turn to black & white. . .
    Igray = imread(name);
    Ithresh = Igray > 175;
    BW = imcomplement(Ithresh);

    % Get rid of small amounts of sand . . .
    % Note: maybe don't need this
    SE = strel('disk', 2, 8);
    BW = imdilate(BW, SE);

end


% The post code (in our case) is the index - 1 . . .
postCode = maxIndex - 1


postCode =

     6     0     3     0     2     3
```

Published with MATLAB® R2017a