Table of Contents

This code evaluates the test set.

```
% ** Important.  This script requires that:
% 1)'centroid_labels' be established in the workspace
% AND
% 2)'centroids' be established in the workspace
% AND
% 3)'test' be established in the workspace

% You should save 1) and 2) in a file named 'classifierdata.mat' as
 part of
% your submission.

centroid_labels = centroids(:, 785);
predictions = zeros(200,1);
outliers = zeros(200,1);


% loop through the test set, figure out the predicted number
for i = 1:size(test)

testing_vector=test(i,:);

% Extract the centroid that is closest to the test image
[prediction_index,
 vec_distance]=assign_vector_to_centroid(testing_vector,centroids);

predictions(i) = centroid_labels(prediction_index);

end
```

DESIGN AND IMPLEMENT A STRATEGY TO SET THE outliers VECTOR

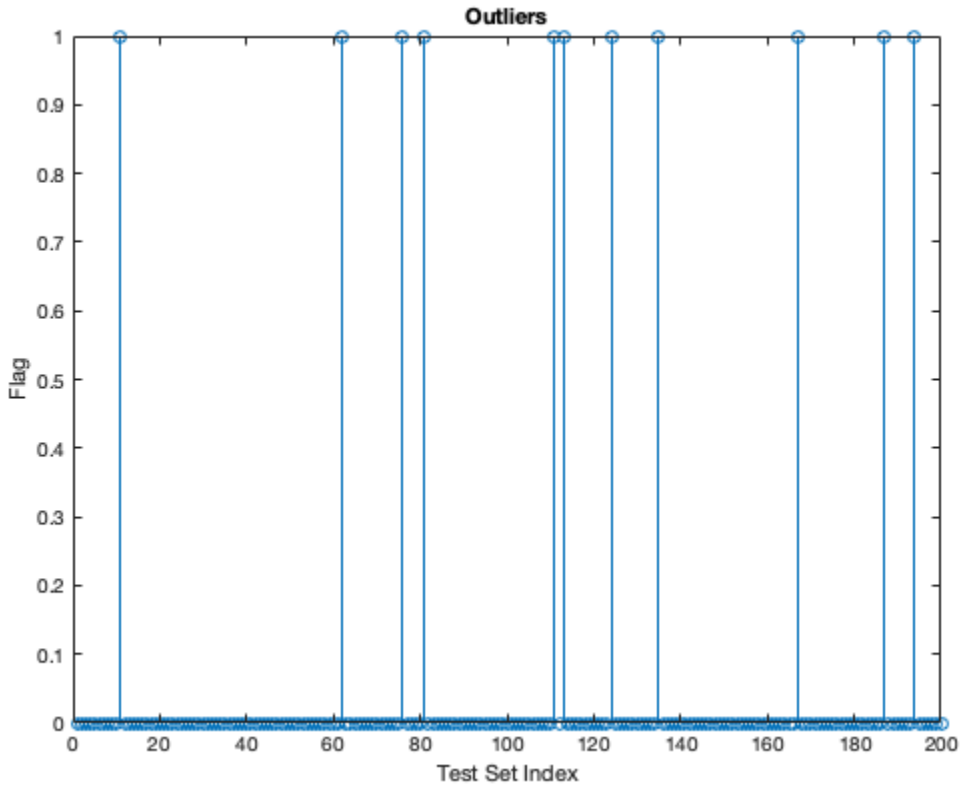
outliers(i) should be set to 1 if the i^th entry is an outlier otherwise, outliers(i) should be 0

```
for i=1:size(test)
        if test(i, 1) > 0
            outliers(i,1)= 1;
        end
```

```
end
```

MAKE A STEM PLOT OF THE OUTLIER FLAG

```
figure;
stem(outliers);
title('Outliers');
xlabel('Test Set Index');
ylabel('Flag');
```

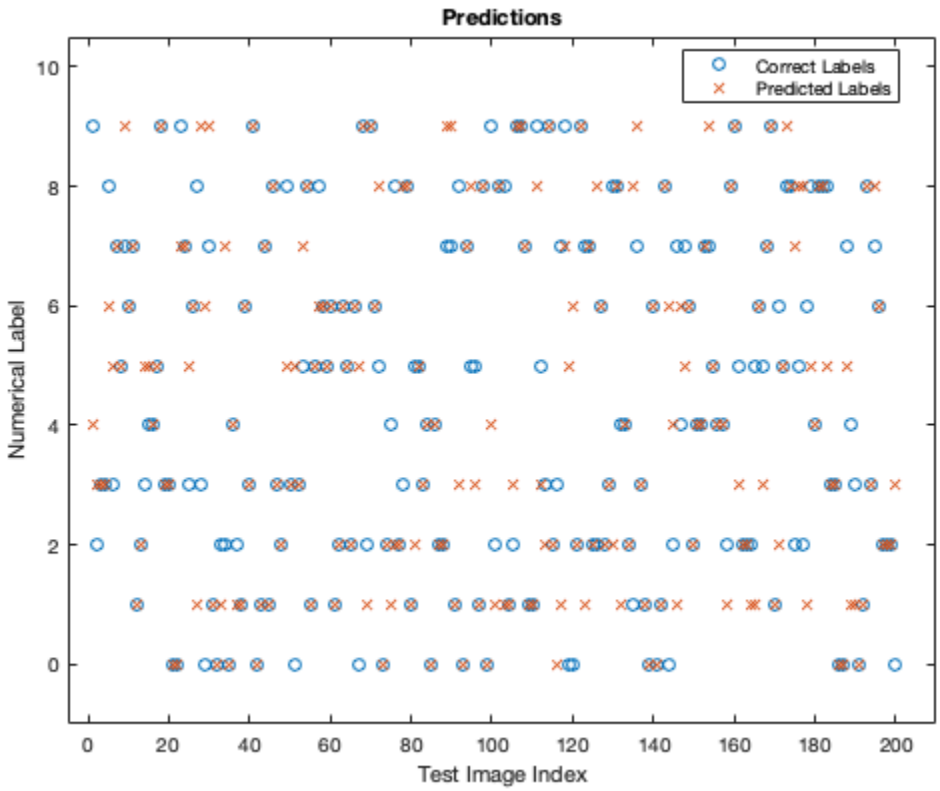


The following plots the correct and incorrect predictions

Make sure you understand how this plot is constructed

```
figure;
plot(correctlabels,'o');
hold on;
plot(predictions,'x');
title('Predictions');
xlabel('Test Image Index');
ylabel('Numerical Label');
xlim([-5,210]);
ylim([-1,10.5]);
```

```
legend('Correct Labels','Predicted Labels','Location', 'best');
```



The following line provides the number of instances where and entry in correctlabel is

equatl to the corresponding entry in prediction However, remember that some of these are outliers

```
sum(correctlabels==predictions);
disp('Number of correct predictions: ');
disp(sum(correctlabels==predictions));
disp(' ');
disp('Percentage of test images correctly assigned: ');
disp(sum(correctlabels==predictions)/length(predictions)*100);

function [index, vec_distance] =
 assign_vector_to_centroid(data,centroids)
    k = size(centroids, 1);
    distances = zeros(k,1);
    values = 1:k;

    for cenIn=1:k
        distances(cenIn)= norm(data(2:length(data)) -
 centroids(cenIn,2:size(centroids,2)));
    end
```

```
    % return index as the centroid number
     index = values(distances == min(distances));

     % return vec_distance as the minimum distance
     vec_distance = min(distances);

end
```

```
Number of correct predictions:
   126


Percentage of test images correctly assigned:
    63
```

Published with MATLAB® R2019a