

```

                                MNIST_Abstraction
%=====
%LOADS MNIST DATA
%=====

clear data_matrix

N = 11^2;

counter = 1;

num_images = 100;

k = 9;

avg_num_points = 0;

min_points = Inf;

directory = ['C:/Users/erdosfan/Desktop/Image_Research/MNIST2/MNIST Dataset JPG
format/MNIST - JPG - training/' int2str(k) '/'];

for i = 1 : num_images

I = imread([directory '1 (' int2str(i) ').jpg']);

[X Z] = read_MNIST(I);

object_array{i} = [X' Z'];

num_points = size(X,2);

avg_num_points = avg_num_points + num_points;

endfor

avg_num_points = avg_num_points / num_images

%=====
%REMOVES OUTLIER-SIZED OBJECTS
%=====

avg_num_points = floor(avg_num_points);

data_matrix = [];
counter = 1;

for i = 1 : num_images

```

```

                                MNIST_Abstraction

temp = object_array{i};
num_points = size(temp,1);

if(num_points >= avg_num_points)

data_matrix(:, :, counter) = temp(1:avg_num_points, :);

temp_x = data_matrix(1:avg_num_points, 1, counter);
temp_y = data_matrix(1:avg_num_points, 2, counter);

data_matrix_cat(counter, :) = [temp_x' temp_y'];
counter = counter + 1;

endif

endfor

%=====
%GENERATES DELTA
%=====

data_matrix_cat(:, 2*avg_num_points + 1) = -1; %this is a dummy classifier

[final_data_categories_array final_delta] =
optimize_categories_fast_N(data_matrix_cat, 2*avg_num_points);

%=====
%CALCULATES ABSTRACTION
%=====

i = 1;

num_items = size(data_matrix_cat, 1);

while(num_items > 1)

[max_matches final_abstraction final_category match_list] =
MNIST_generate_abstraction(data_matrix, data_matrix_cat, final_delta,
avg_num_points);

data_matrix(:, :, match_list) = [];
data_matrix_cat(match_list, :) = [];

data_categories_array{i} = final_category; %stores the category generated by each
iteration

i = i + 1;
num_items = size(data_matrix_cat, 1)

```

MNIST_Abstraction

```
size(data_matrix)

endwhile

if(num_items == 1)

data_categories_array{i} = data_matrix_cat(1,:);

endif
```