

```

                                IMG_Comparison
function average_score = IMG_Comparison(region_matrix_1, N_1, S_mat_1,
score_vector_1, region_matrix_2, N_2, S_mat_2, score_vector_2)

if(entropy(region_matrix_1) > entropy(region_matrix_2)) % the image with more
detailed structure is treated as the input image

    region_matrix_input = region_matrix_1;

    N_input = N_1;

    S_mat_input = S_mat_1;

    score_vector_input = score_vector_1;

    region_matrix_comp = region_matrix_2;

    N_comp = N_2;

    S_mat_comp = S_mat_2;

    score_vector_comp = score_vector_2;
else
    region_matrix_input = region_matrix_2;

    N_input = N_2;

    S_mat_input = S_mat_2;

    score_vector_input = score_vector_2;

    region_matrix_comp = region_matrix_1;

    N_comp = N_1;

    S_mat_comp = S_mat_1;

    score_vector_comp = score_vector_1;
endif

weight_vector = generate_weight_matrix(N_input, region_matrix_input, S_mat_input);

average_score = 0;
divisor = sum(weight_vector);

```

```

                                IMG_Comparison
num_features = max(region_matrix_input(:));
num_regions = max(region_matrix_comp(:));
is_taken = zeros(1,num_regions);

[max_weight k] = max(weight_vector);

while(max_weight > 0)

    min_score = norm(score_vector_input{k});

    for i = 1 : num_regions

        if(is_taken(i) == 0 && (norm(score_vector_input{k} - score_vector_comp{i}) <
min_score))

            closest_region = i;
            min_score = norm(score_vector_input{k} - score_vector_comp{i});
            score_vector_input{k};
            score_vector_comp{i};

        endif

    endfor

    average_score = average_score + weight_vector(k)*min_score;
    is_taken(closest_region) = 1;
    weight_vector(k) = 0;
    [max_weight k] = max(weight_vector);

endwhile

average_score = average_score/divisor;

endfunction

```