

```

                                score_image
function [ent_vector, avg_color_vector, std_dev_color_vector, std_dev_inf_vector] =
score_image(input_im,N,avg_ent,s_0)

im_size = size(input_im);
num_rows = im_size(1);
num_cols = im_size(2);
row_length = floor(num_rows/N);
col_length = floor(num_cols/N);
ent_vector = 0;
avg_color_vector = 0;
std_dev_color_vector = 0;
std_dev_inf_vector = 0;
for i = 0 : N - 1
    for j = 0 : N - 1
temp_im = input_im(1 + i*row_length : (i+1)*row_length, 1 + j*col_length :
(j+1)*col_length, 1 : 3);
temp_ent = entropy(temp_im);
ent_vector = [ent_vector entropy(temp_im)];
temp_avg_color = mean(temp_im(:));
avg_color_vector = [avg_color_vector temp_avg_color];
temp_std_dev = std(temp_im(:));
std_dev_color_vector = [std_dev_color_vector temp_std_dev];
temp_std_dev_inf = (temp_ent - avg_ent)^2/s_0^2;
std_dev_inf_vector = [std_dev_inf_vector temp_std_dev_inf];
    endfor
endfor

```

```
                                score_image
ent_vector = ent_vector(2:N^2 + 1);
avg_color_vector = avg_color_vector(2:N^2 + 1);
std_dev_color_vector = std_dev_color_vector(2:N^2 + 1);
std_dev_inf_vector = std_dev_inf_vector(2:N^2 + 1);
endfunction
```