

```
function [total_diff avg_matrix] = calculate_total_im_diff(indexes, img_matrix, N)
```

```
im_rows = size(img_matrix,1);
```

```
im_cols = size(img_matrix,2);
```

```
num_rows = floor(im_rows / N);
```

```
num_cols = floor(im_cols / N);
```

```
num_partition_rows = N; %the number of resultant partition rows
```

```
num_partition_cols = N; %the number of resultant partition cols
```

```
pixels_per_cell = num_rows*num_cols;
```

```
num_pixels = size(indexes,1); %total number of pixels in the partitioned image
```

```
num_partition_cells = num_pixels / pixels_per_cell;
```

```
pixel_entries = img_matrix(indexes);
```

```
avg_matrix = reshape(pixel_entries, [1 pixels_per_cell num_partition_cells]);
```

```
avg_matrix = mean(avg_matrix);
```

```
avg_matrix = reshape(avg_matrix,[num_partition_rows num_partition_cols]);
```

```
%-----
```

```
%compares the matrixes
```

```
%up / down
```

```
temp_matrix = zeros(num_partition_rows,num_partition_cols);
```

```
temp_matrix(1 : num_partition_rows - 1, :) = avg_matrix(2 : num_partition_rows, :);
```

```
UD_diff = abs(avg_matrix .- temp_matrix);
```

```
UD_diff(num_partition_rows,:) = 0; %this ignores the bottom of the image;
```

```
UD_diff = sum(UD_diff(:));
```

```
%left / right
```

```
temp_matrix = zeros(num_partition_rows,num_partition_cols);
```

```
temp_matrix(:, 1 : num_partition_cols - 1) = avg_matrix(:, 2 : num_partition_cols);
```

```
LR_diff = abs(avg_matrix .- temp_matrix);
```

```
LR_diff(:,num_partition_cols) = 0; %this ignores the bottom of the image;
```

```
LR_diff = sum(LR_diff(:));
```

```
total_diff = UD_diff + LR_diff;
```

```
endfunction
```