You have to develop an application able to analyze an overhead view of a parking lot and determine the number of free spots and their location inside of the image. In particular, you assume that each parking spot have the same pixel size (50x25) and the image is perfectly aligned such the one reported below (the top left corner of the first parking area coincides with the one of the input image).

Write a MATLAB script able to perform the following steps:

a) Read the 8-bit input color image (stored in the file *'parking_lot.png'*), using a double representation, and visualize it.
b) Convert the input image to grayscale and apply a gaussian filter followed by an isotropic Laplacian filter using convolutions in order to obtain *img_log*. Use 0.5 as the gaussian standard deviation and a 3x3 kernel for both operations. Assume the image to be symmetric. *(hint: you can use a single convolution).*
c) Obtain a binary image called *mask* setting to true all the pixel locations in which the absolute value of *img_log* is greater than the 10% of the maximum absolute value stored in *img_log*.
d) Apply a 7x7 median filter to *mask*.
e) In order to count the number of free parking spots and produce a visualization, initialize a counter called *free_count* to zero and a rgb image with the same size of the input image called *free_vis* to zero.
f) Consider each 50x25 parking area in the image and perform the following operations: (*hint: you can use: for i = start:step:end*)
    I. Count the number of true values in the corresponding region of *mask.*
    II. Estimate if the considered area is free or occupied by comparing the obtained value to 30% of the considered pixel area.
    III. If the area is free increase *free_count* to one and assign green color to the corresponding region of *free_vis*. Otherwise assign red color to the same area.
g) Perform a linear combination between the color input image and *free_vis* using 0.7 and 0.3 as the corresponding weights. Visualize the result.

```
close all
clear all
clc

%a)
img = im2double(imread('parking_lot.png'));
figure
imshow(img)


%b)
img_gray = rgb2gray(img);
H = fspecial('log',3,0.5);
img_log = imfilter(img_gray,H,'symmetric','conv');

%c)
img_log=abs(img_log);
mask = im2bw(img_log,0.05*max(img_log(:)));

%d)
mask=medfilt2(mask,[7,7]);

%e)
free_count = 0;
free_vis = zeros(size(img));

%f)
p_h=50;
p_w=25;
thr=p_h*p_w*0.3;
for i = 1:p_w:size(img,2)
    for j = 1:p_h:size(img,1)
        img_crop = mask(j:(j+p_h-1),i:(i+p_w-1));
        %f.1)
        count = sum(img_crop(:));
        %f.2)
        if(count<thr)
           %f.3)
           free_count = free_count+1;
           free_vis(j:(j+p_h-1),i:(i+p_w-1),1) = 0;
           free_vis(j:(j+p_h-1),i:(i+p_w-1),2) = 1;
           free_vis(j:(j+p_h-1),i:(i+p_w-1),3) = 0;
        else
           free_vis(j:(j+p_h-1),i:(i+p_w-1),1) = 1;
           free_vis(j:(j+p_h-1),i:(i+p_w-1),2) = 0;
           free_vis(j:(j+p_h-1),i:(i+p_w-1),3) = 0;
        end
    end
end

%g)
figure
imshow(0.3*free_vis+0.7*img)
```