ECE 558 - Project 3

- Kaushik Pillalamarri (200483260)
Sushanth Chilla (200483019)
Tanisha Khurana (200483139)

Blob detection Algorithm

In this project we have implemented a Laplacian of Gaussian Blob detection algorithm. The idea is to involve the image with a Laplacian of Gaussian filter and convolve it for multiple scales to find the maximum or minimum extreme values.

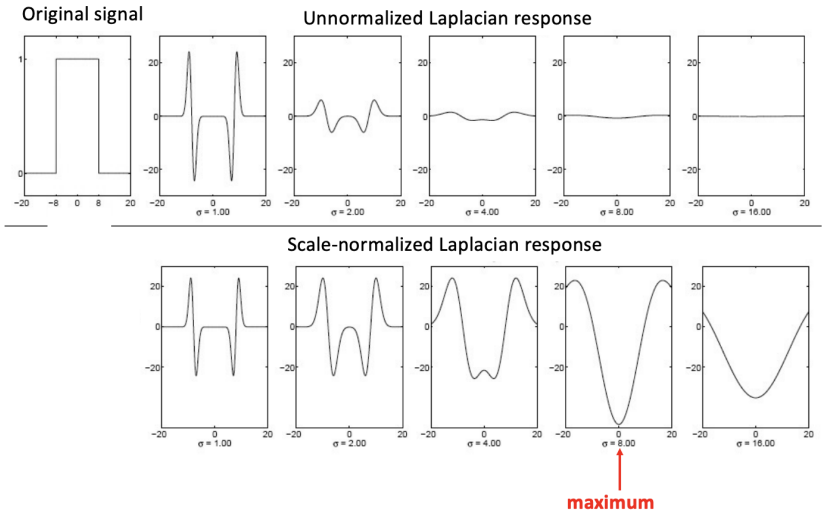
The **algorithm** is as follows:

1. Generate Laplacian of Gaussian kernels with different sigma values.
2. Using our DFT 2d function we get both the kernel and image in the frequency domain. Same padding the kernel and then multiplying them.
3. Computing the scale space with the convolved images of different sigma values. Here, we scaled the sigma with different k values from range 0 to numScales which we give here as 10.
4. Using non-max suppression 2d we use a threshold value to filter out the maxima value.
5. To detect the blobs, we now use 3d non max suppression to get the max value across scale spaces. So, 2 convolved images of adjacent sigma values are compared with one another and if the max value is more than a threshold value we replace the values with the max value to get the local extrema.
6. To check for redundancy we now use a spatial tree to search for pairs of blobs.
7. And calculate the overlap between the distances between 2 blobs. This filters out the redundant blobs.
8. We now save the maxima values in the markers array. These now give us the (r,x,y) coordinates of the extreme values.
9. Iterating through the blob markers we display the blobs onto the image.

The first step involves generating the Laplacian of Gaussian kernels of different scales to detect the sharpened corners, which we use to convolve it with the given image.

$$(x^2 + y^2 - 2\sigma^2)\, e^{-(x^2+y^2)/2\sigma^2}$$

These LoG filters are scale normalized i.e they are multiplied by σ^2 and padded using reflection padding to fix the wrap around error.



The varying sigma LoG filters are shown below:

Since convolution in the time domain is a time consuming process, we transform both the kernel and the image into the frequency domain, and convolve it in the frequency domain which is just a matrix multiplication.

In the above process we can obtain the scale space, and all the convolved images are stored in the format (σ, I), where σ is the scaled standard deviation of the laplacian of gaussian filter, and I contains the result of the convolution with the scaled LoG.

We have used 10 scales here and iterated through the scale space to find the maximum extrema values. To find the extrema, we use the function $non\_max\_suppression$.

This function takes in the scale space as argument and it first applies the non-max suppression across each image to find the maxima and the minima i.e the local extrema. We dilate each image in the scale space and equate it with the undilated

convolved image. Thus, we obtain the sharpened edges and use a threshold for filtering. Thus we get the max value.

We then apply it across the scale space convolved images such that the adjacent two scaled convoluted images are compared with each other and we get the scale space maxima. We get the bool values which indicate whether the value is a max or a min and we multiply this with the scale space image.

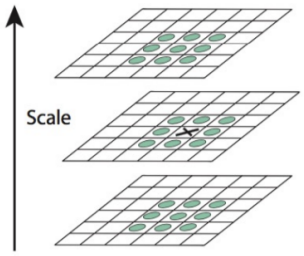


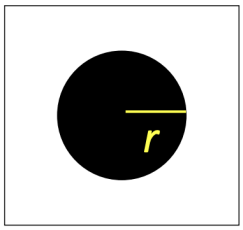
Figure 2: Maxima and minima of the difference-of-Gaussian images are detected by comparing a pixel (marked with X) to its 26 neighbors in 3x3 regions at the current and adjacent scales (marked with circles).

The next step was to find the coordinates of these extreme values. Discarding the first two sigma scale space we used the coordinates of (x,y) as the first 2 values and the 3rd value of the coordinate tuple as the scaled σ .

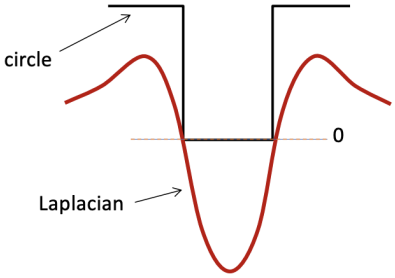
The next step was to filter out the pairs of blobs which had overlapping greater than some finite threshold value.

We used a tree to binary search pairs of blobs to check for overlapping. To filter out the overlapped blobs, we used a function to calculate the distance between the two radii of two overlapped blobs. If the distance was less than the sum of the two radii, we find the area of the overlap. If that area was more than the threshold we set, we discard one blob.

$$\sigma = r / \sqrt{2}.$$

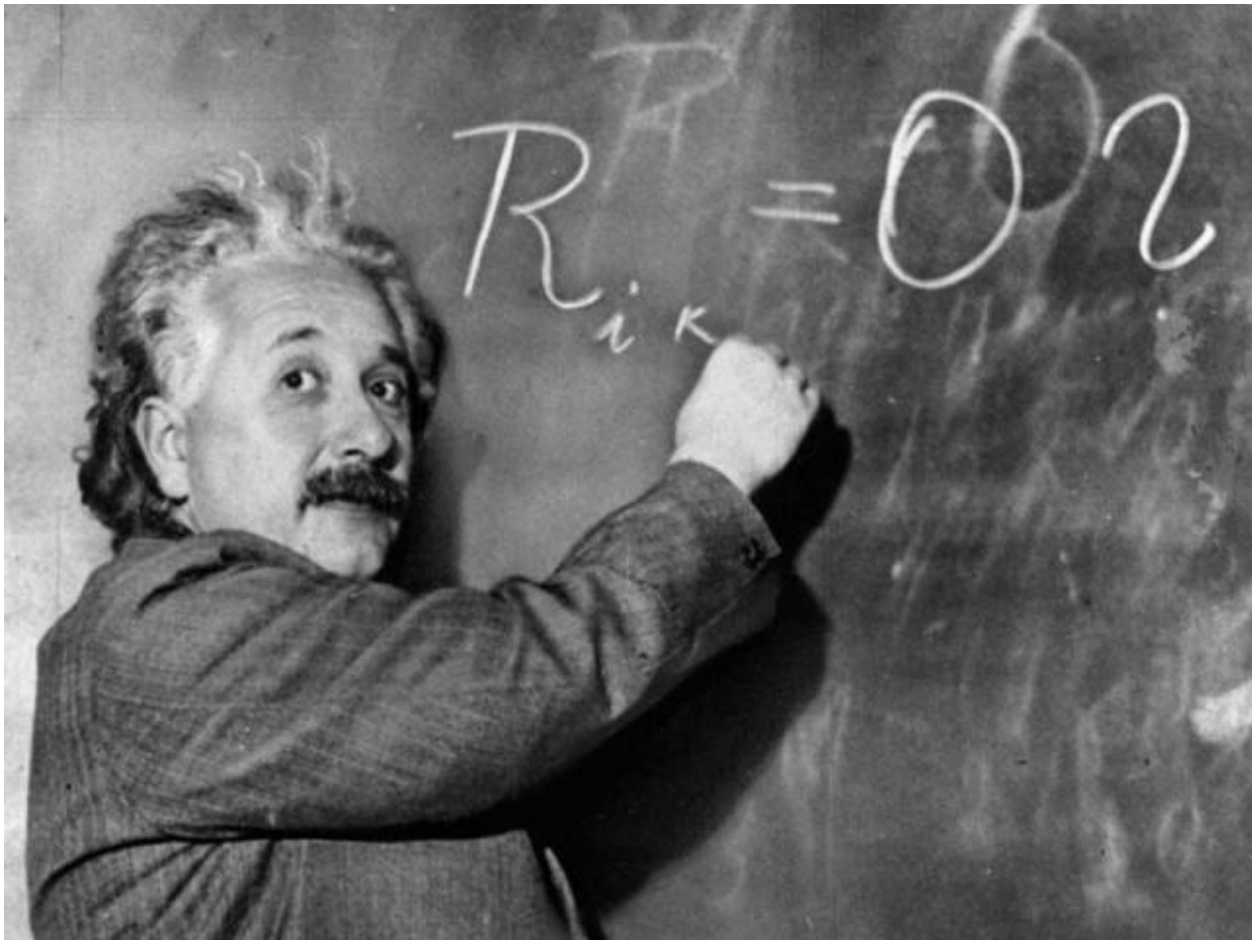


image



We then simply iterate through our coordinates and draw the blobs onto our input image to return the output image.

The input images are shown below:


$R_{ik} = 0$

Team Contributions:
Kaushik was responsible for generating the DoG and LoG filters and the overlap redundancy check, Sushant was responsible for the non-max suppression function and finding coordinates and Tanisha was responsible for the convolution using FFT and for generating the scale space.