

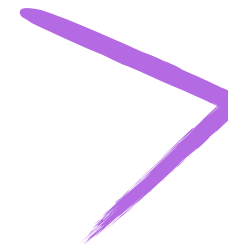
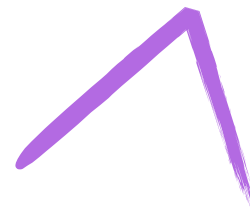
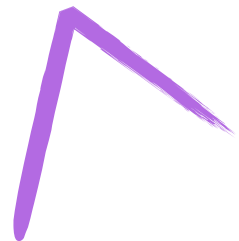


Multi-scale Detection

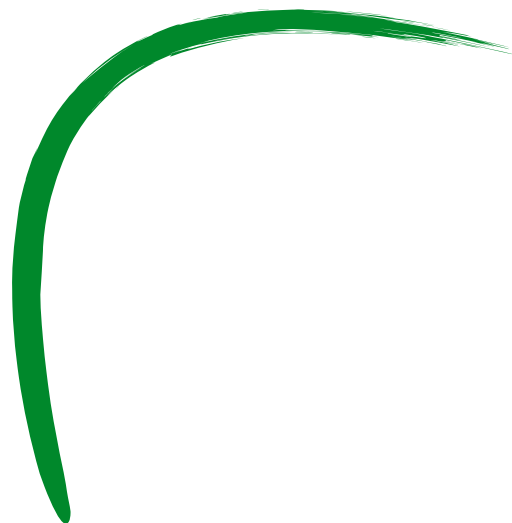
16-385 Computer Vision (Kris Kitani)
Carnegie Mellon University

Properties of the Harris corner detector

Rotation invariant?

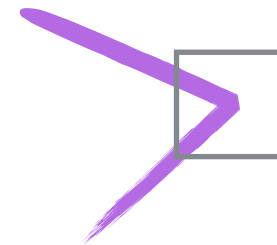
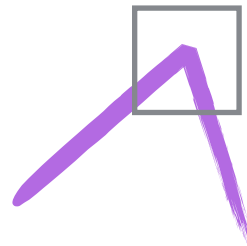
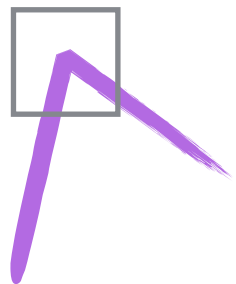


Scale invariant?

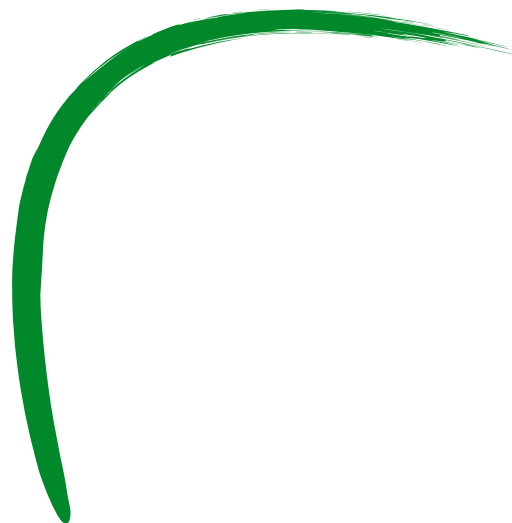


Properties of the Harris corner detector

Rotation invariant?

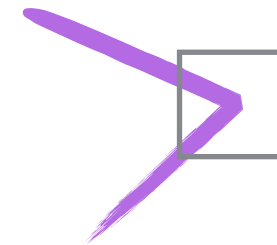
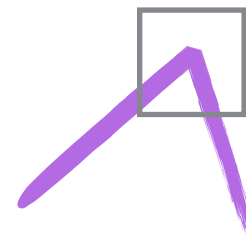
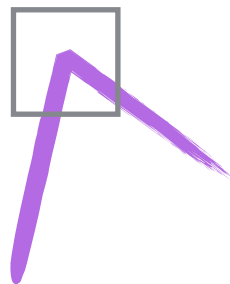


Scale invariant?



Properties of the Harris corner detector

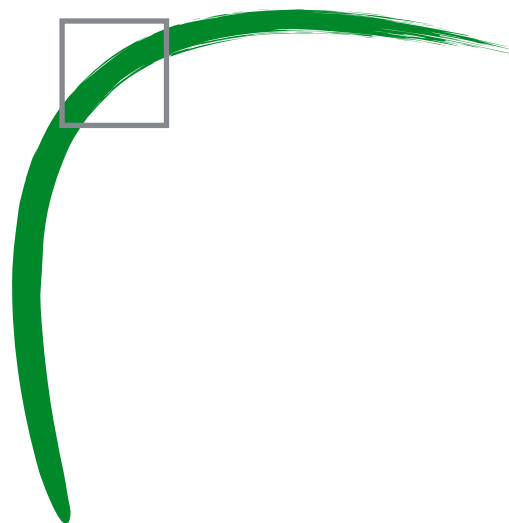
Rotation invariant?



Scale invariant?



edge!



corner!



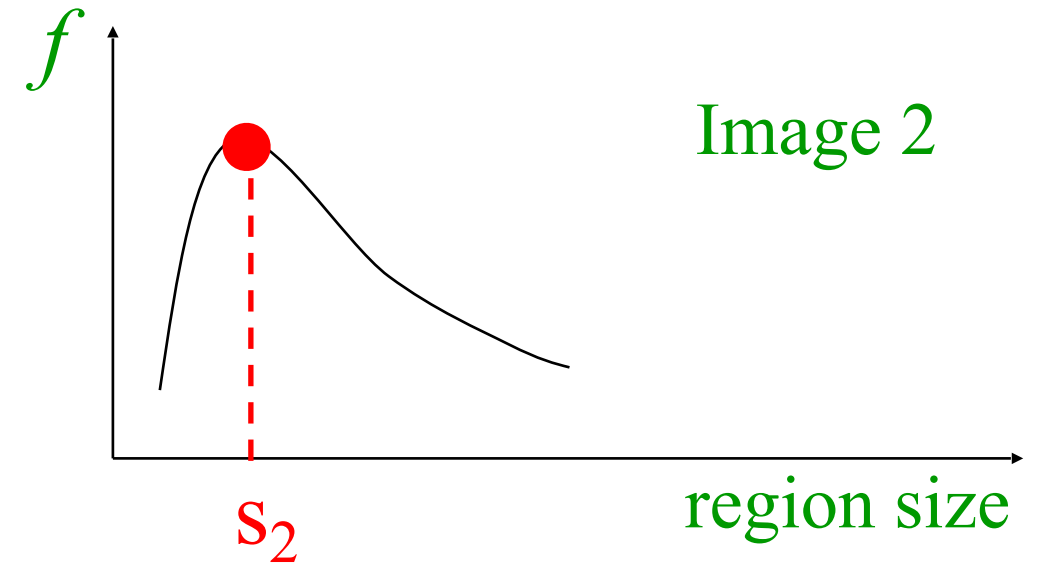
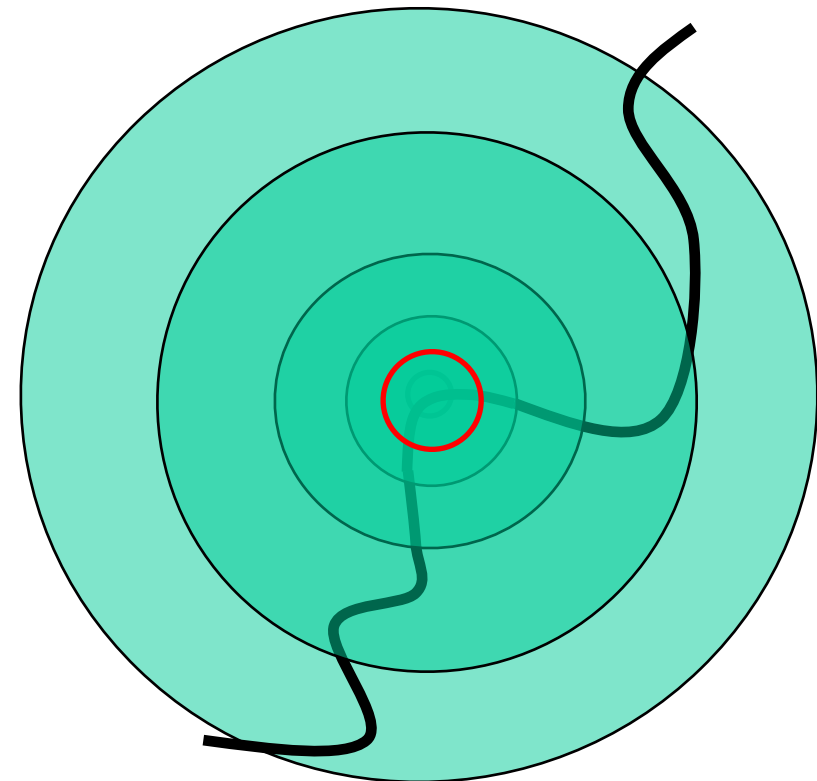
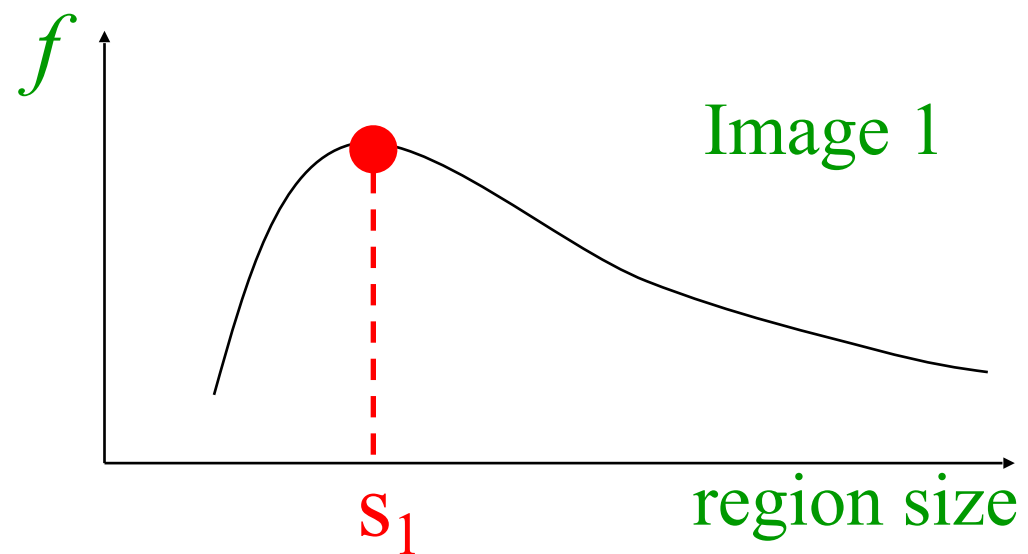
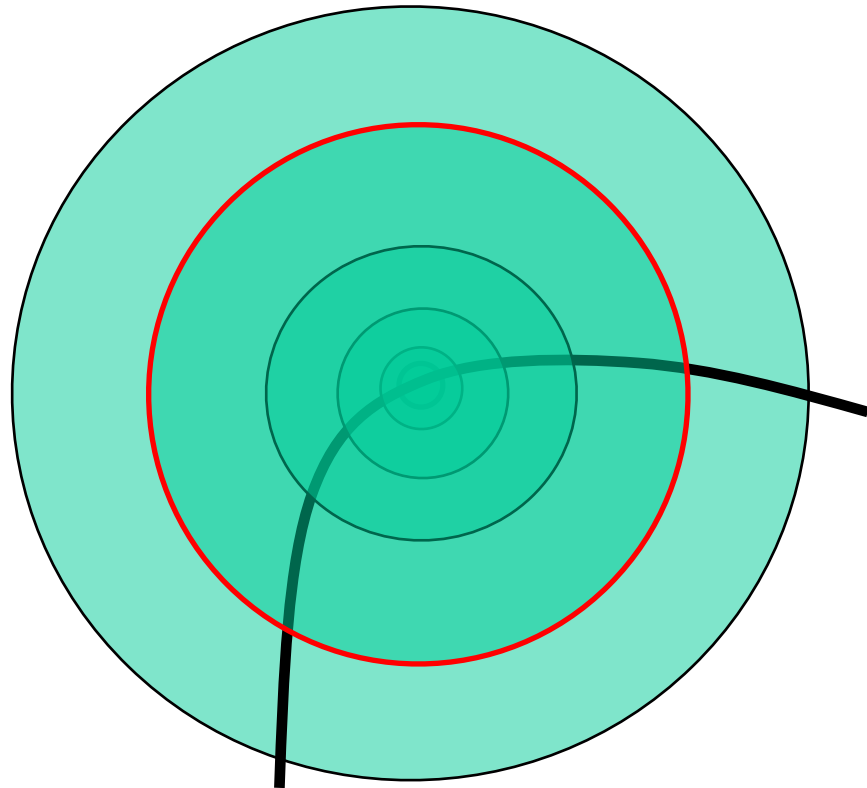
How can we make a feature detector scale-invariant?

How can we automatically select the scale?



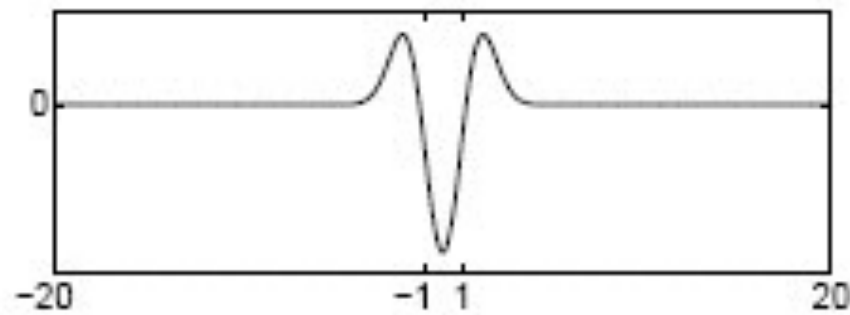
Intuitively...

Find local maxima in both **position** and **scale**

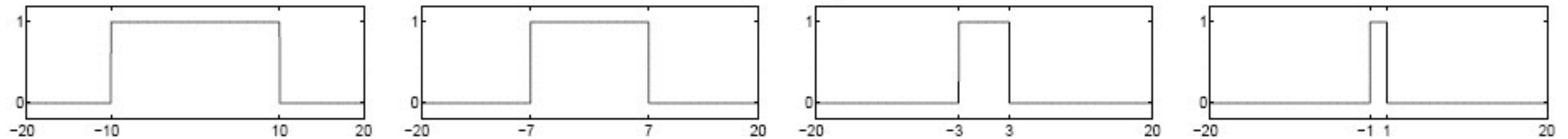


Formally...

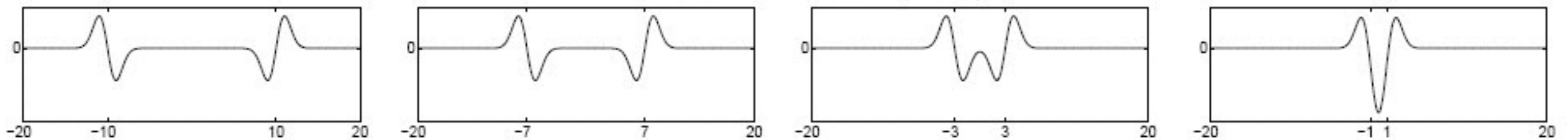
Laplacian filter



Original signal

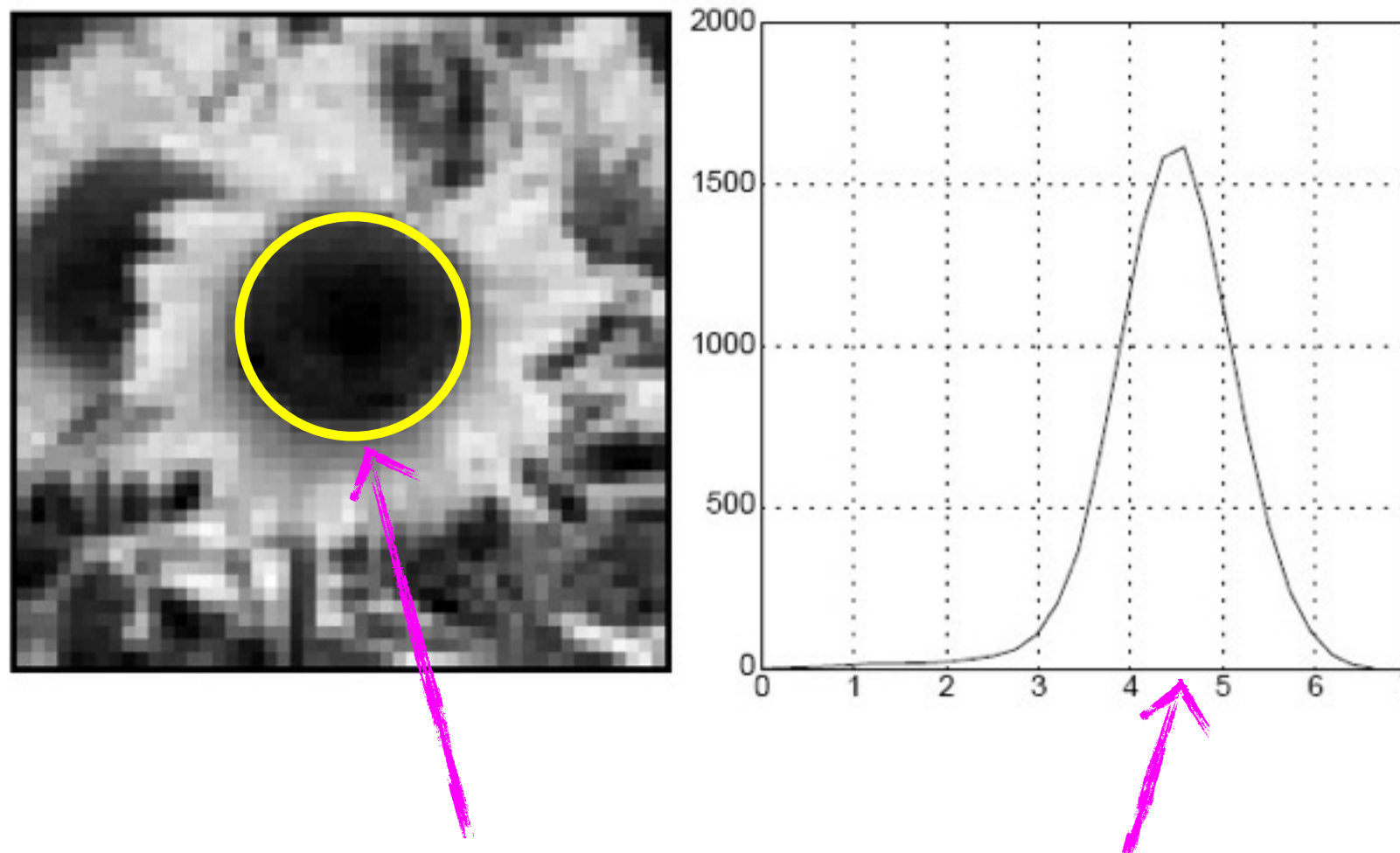


Convolved with Laplacian ($\sigma = 1$)



Highest response when the signal has the same **characteristic scale** as the filter

characteristic scale - the scale that produces peak filter response

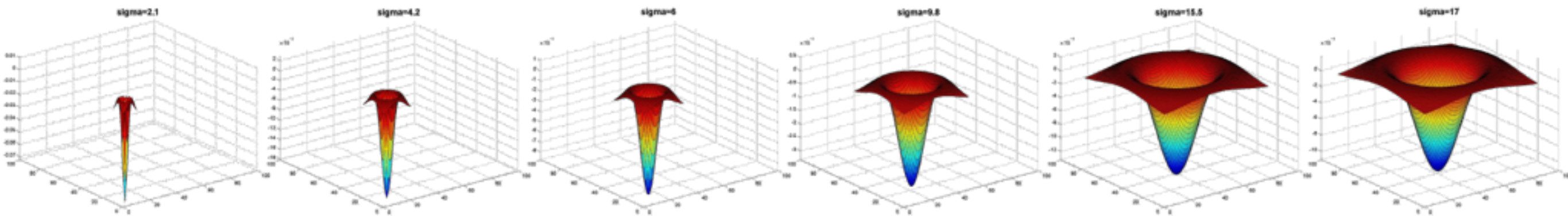


characteristic scale

we need to search over characteristic scales

Multi-scale 2D Blob detection

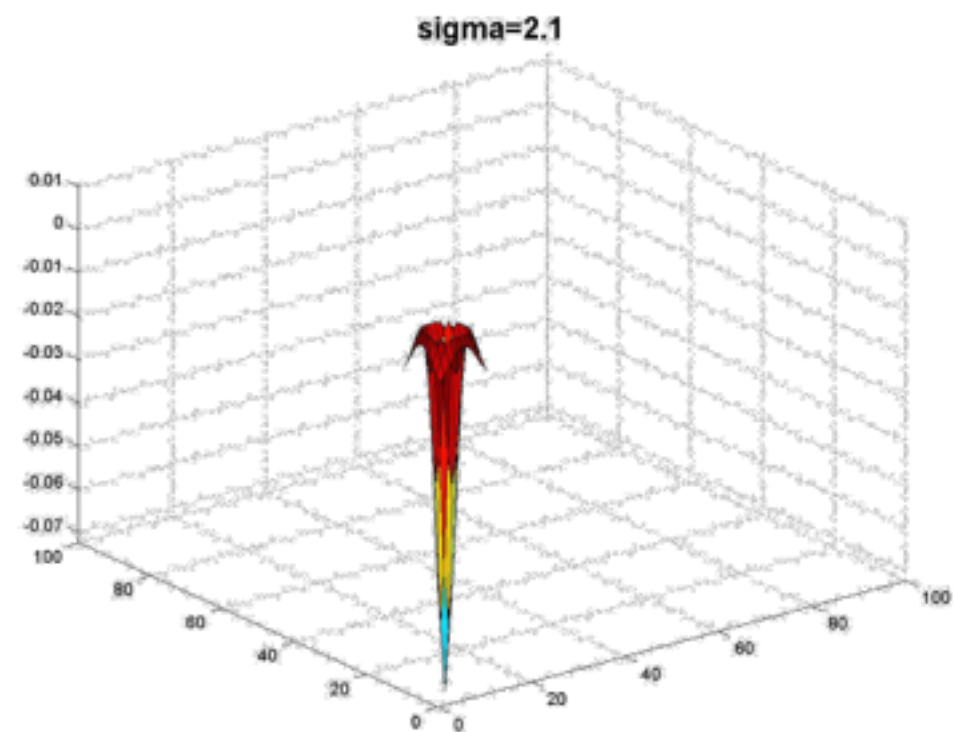
What happens if you apply different Laplacian filters?



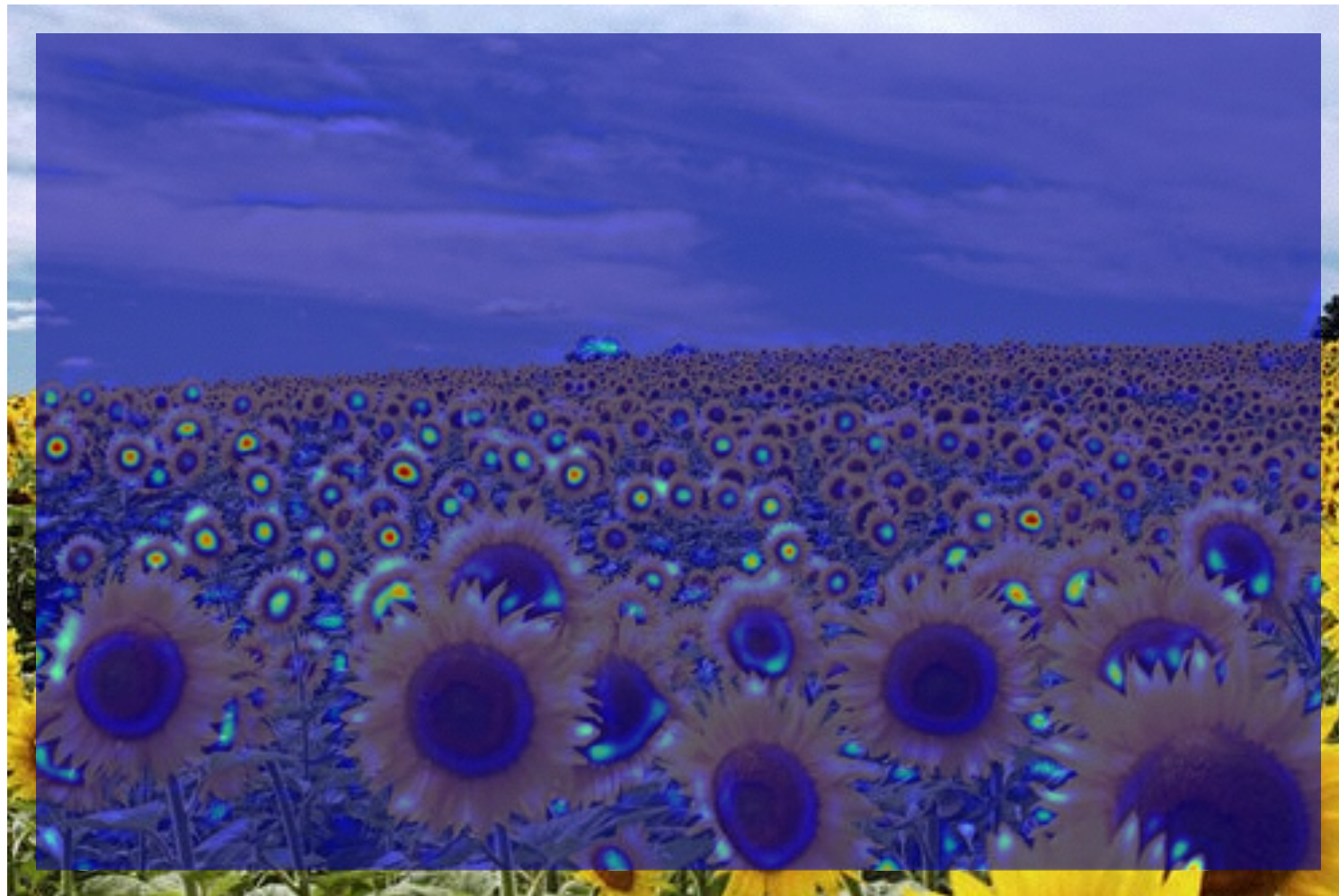
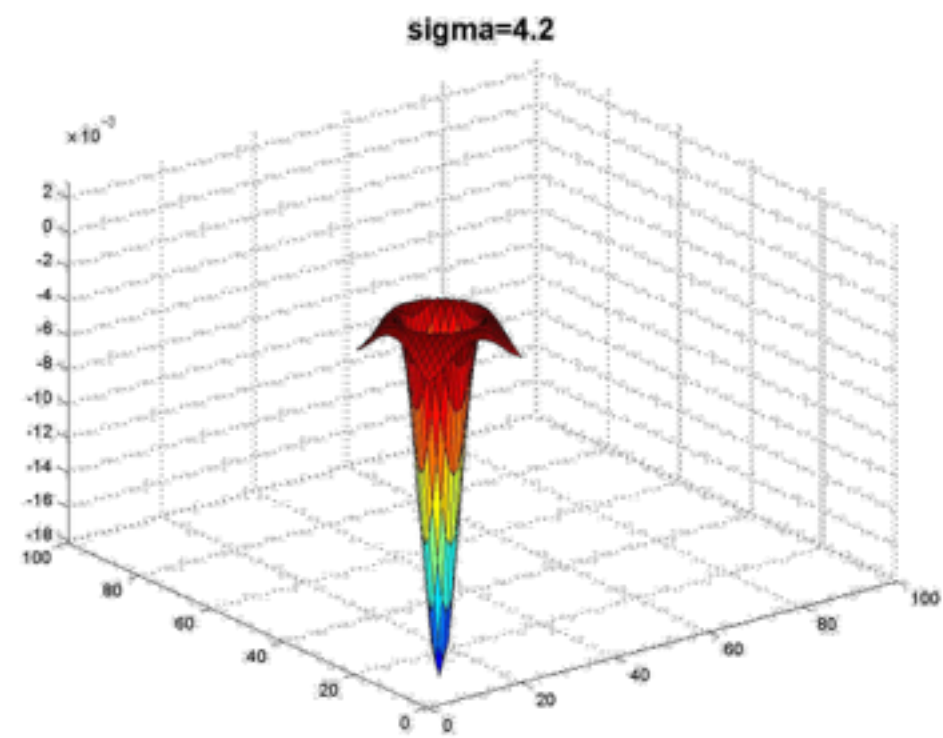
Full size

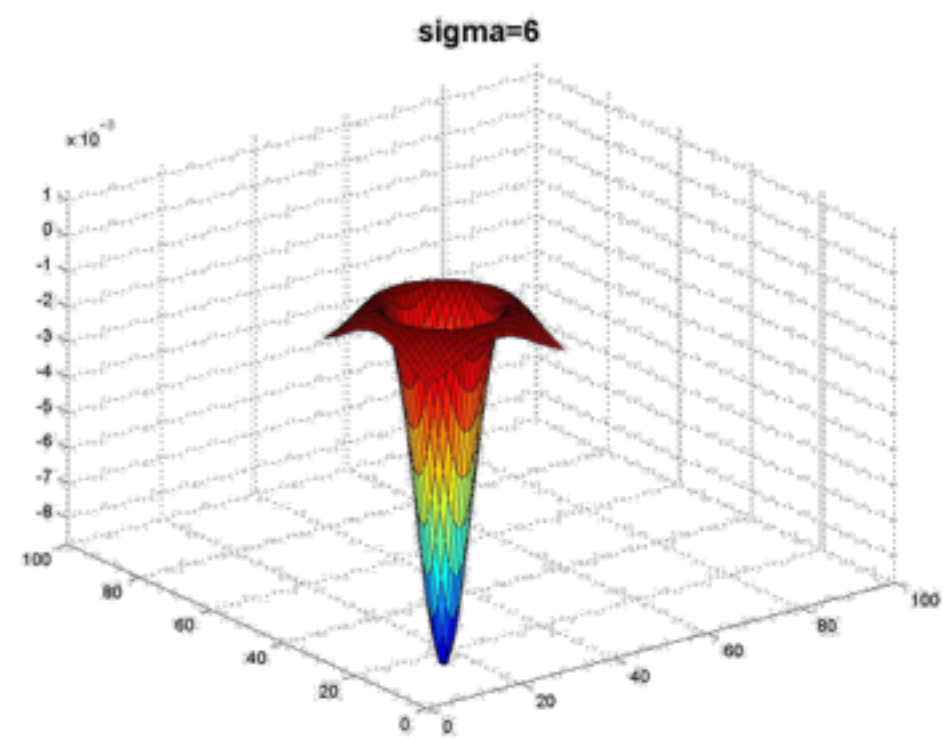
3/4 size

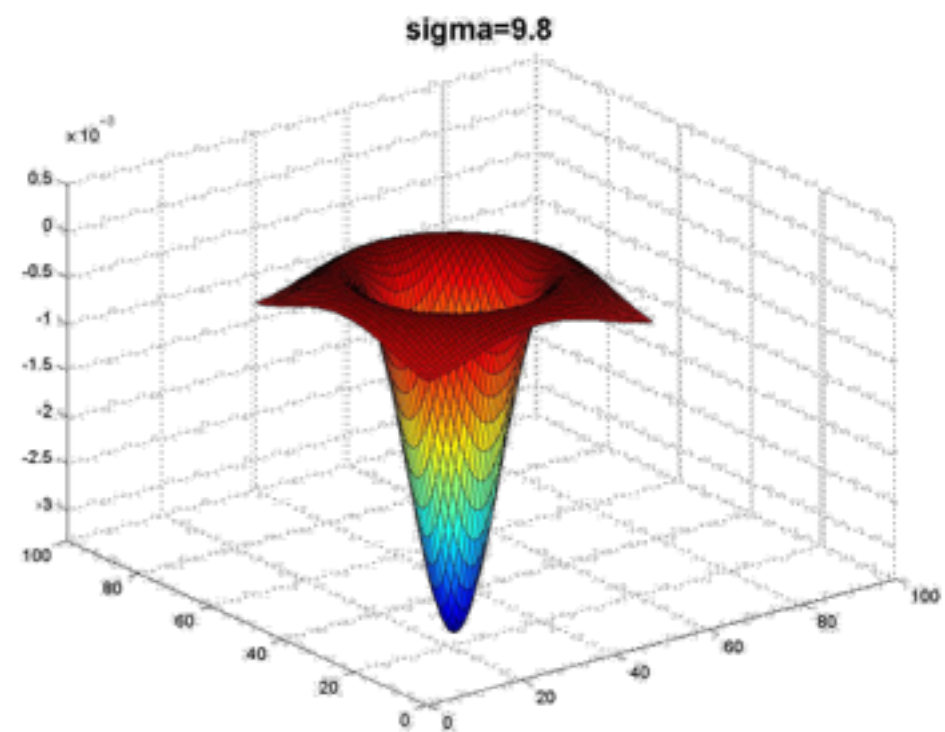


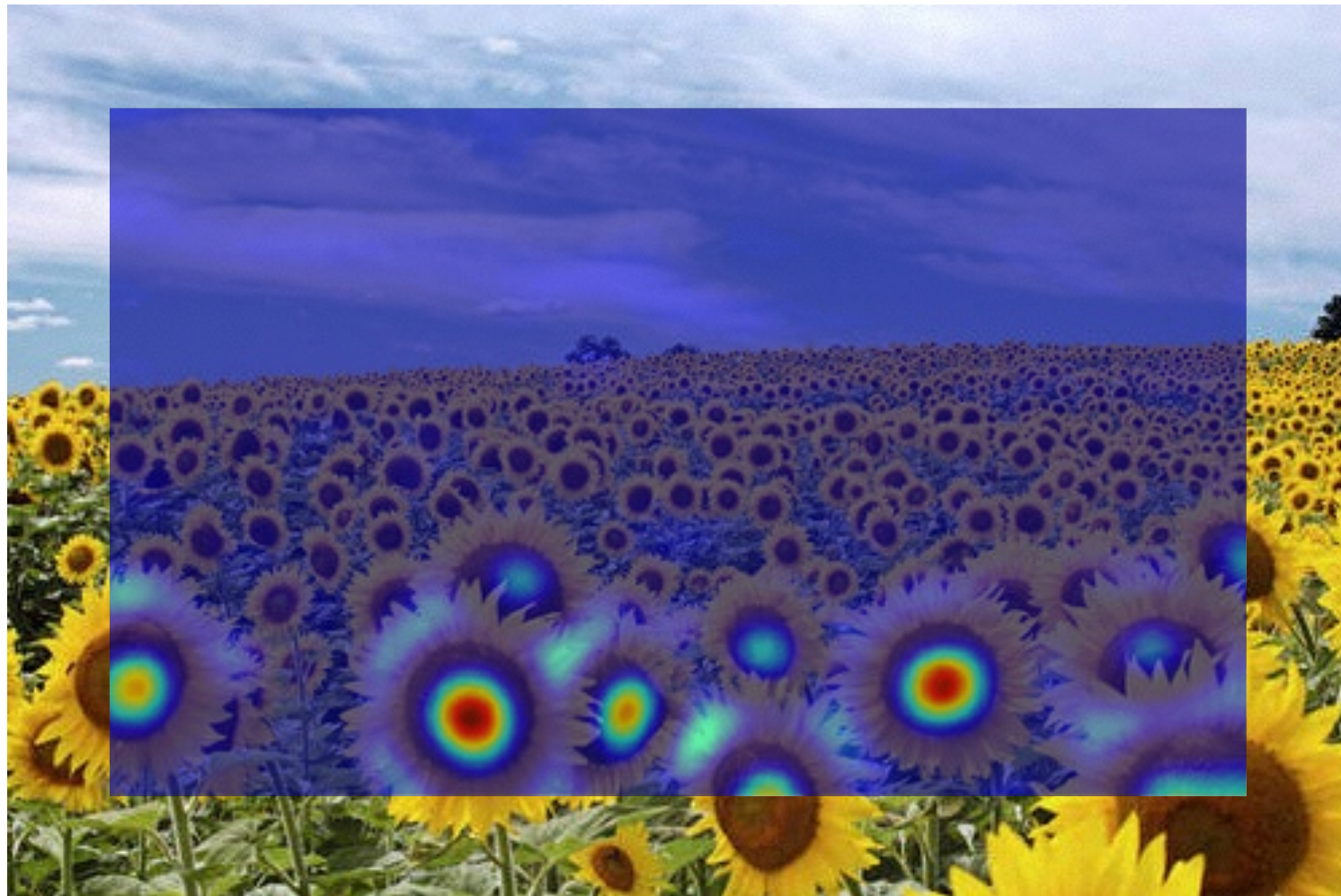
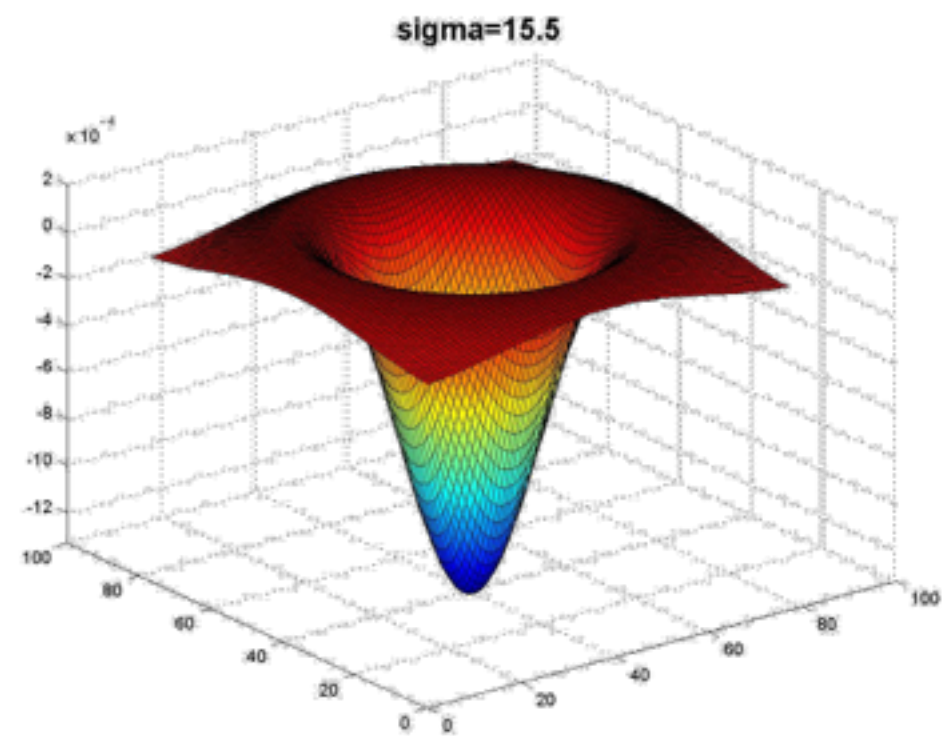


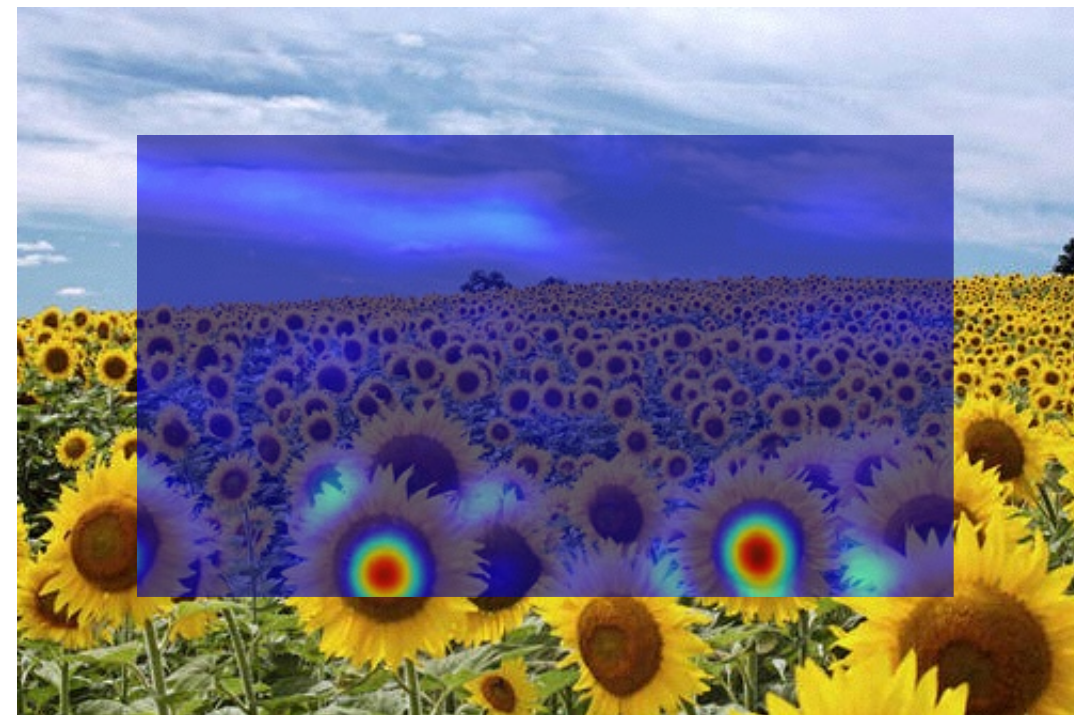
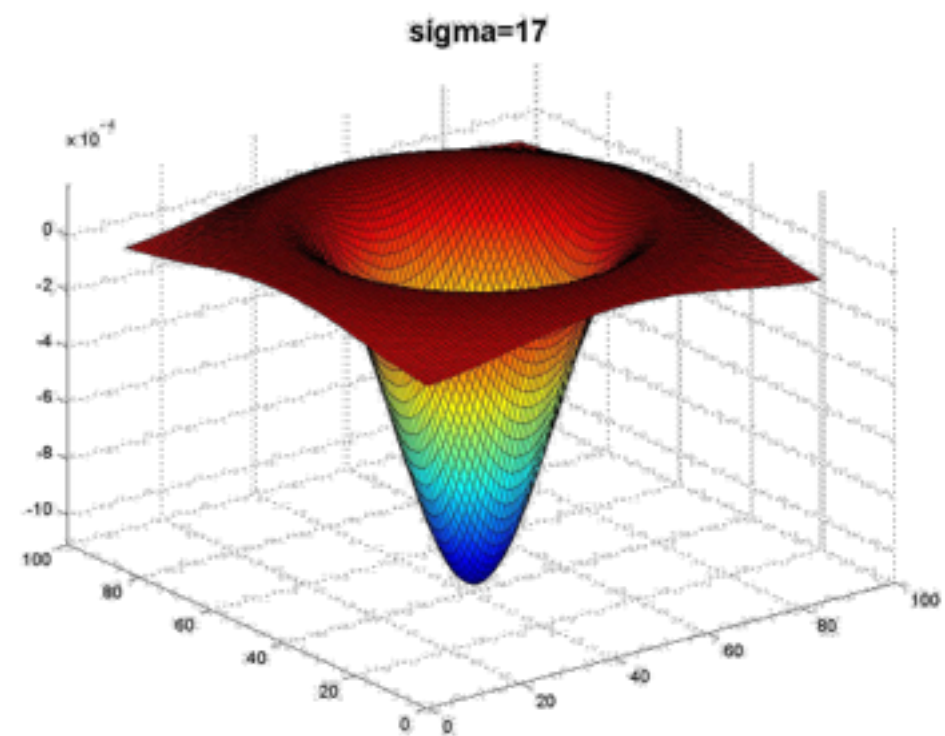
jet color scale
blue: low, red: high











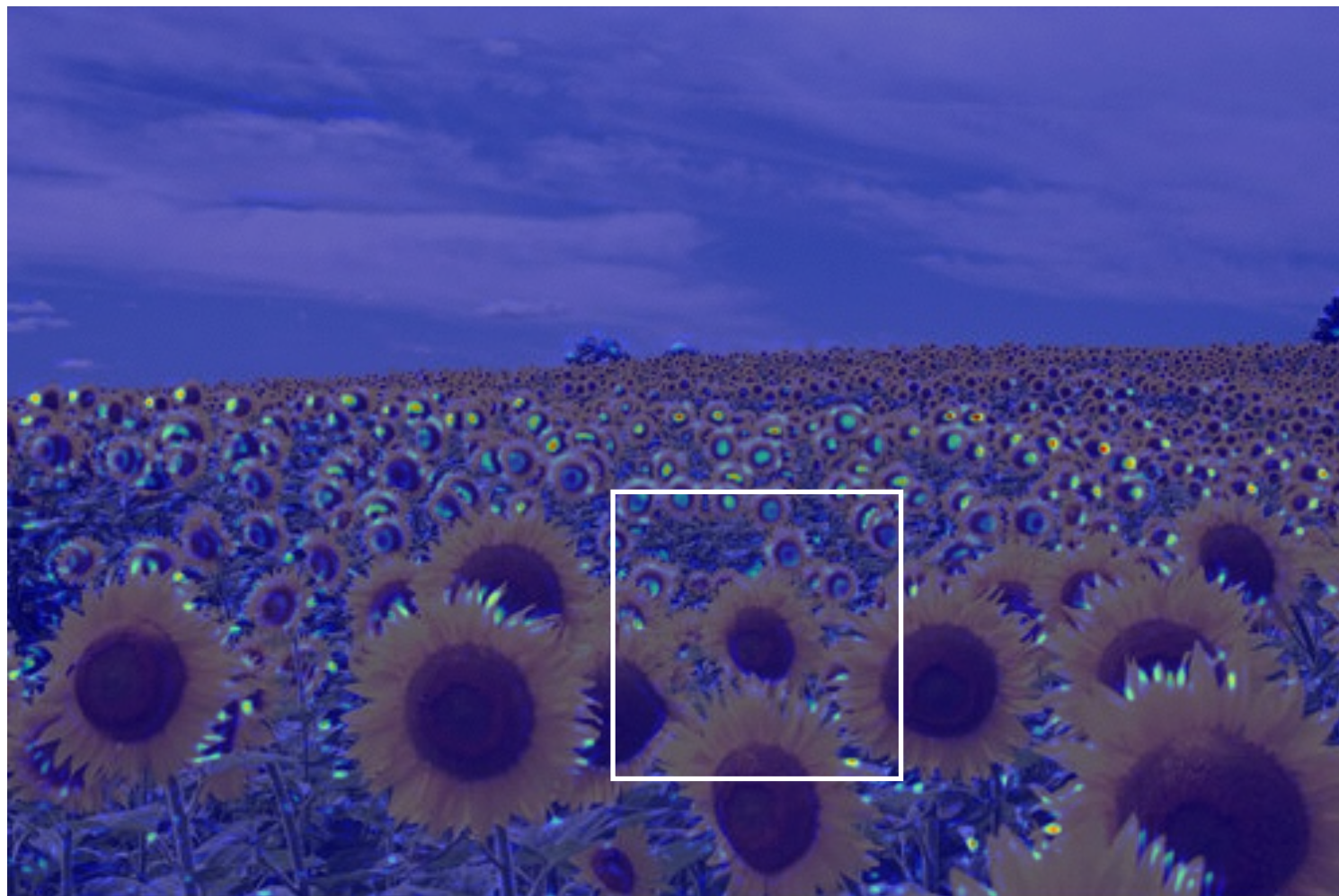
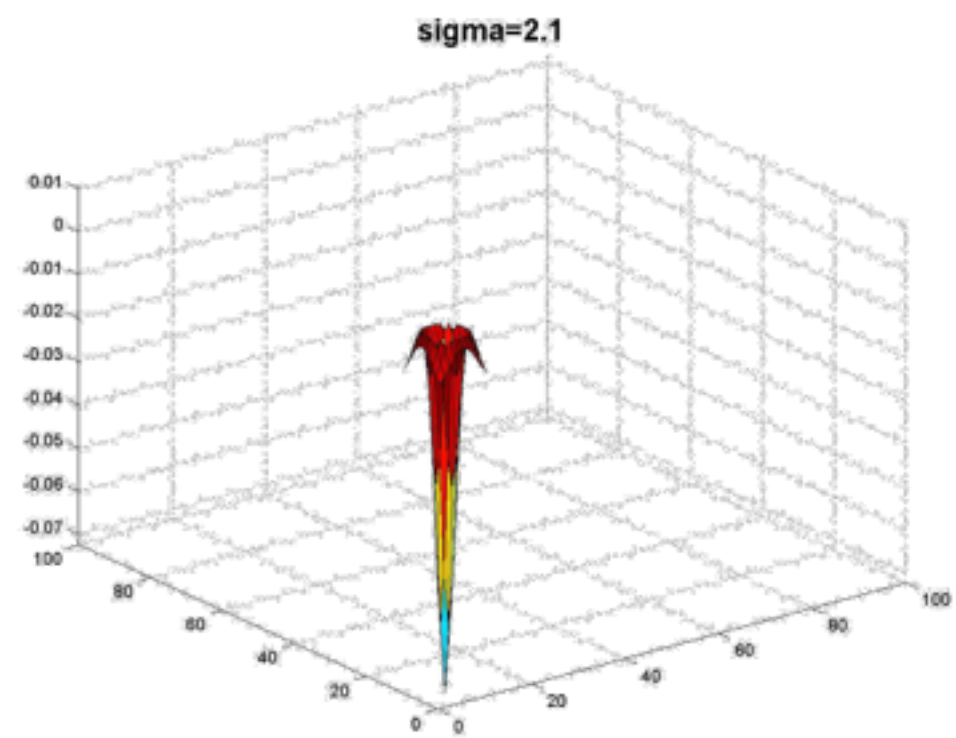
What happened when you applied different Laplacian filters?

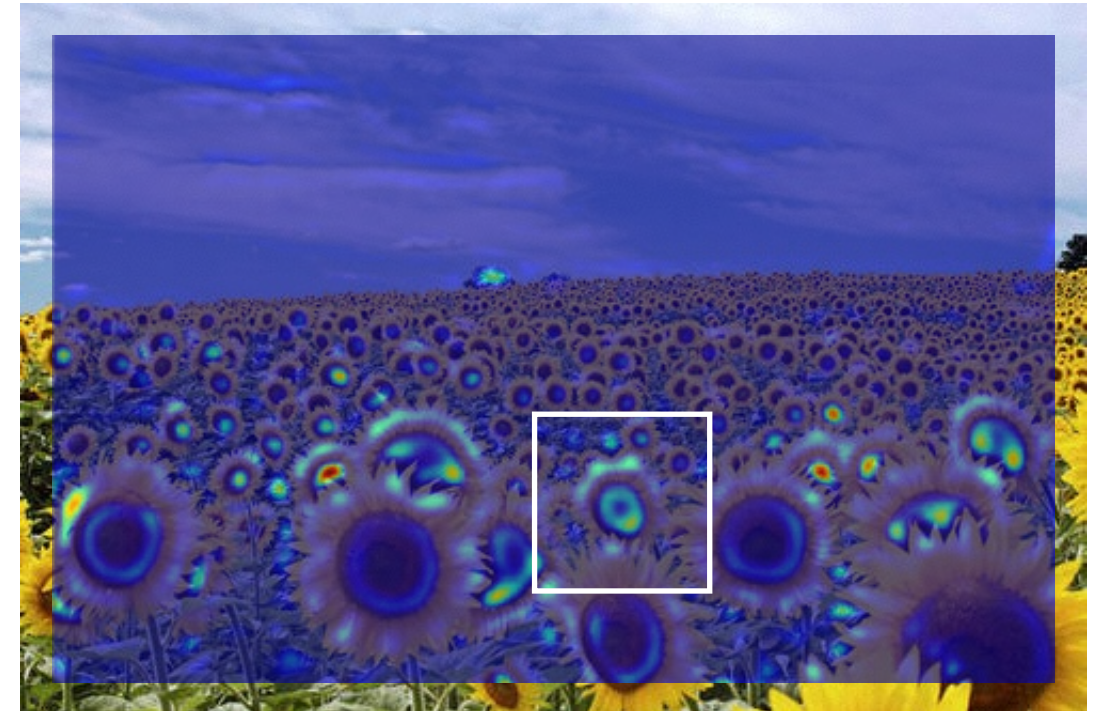
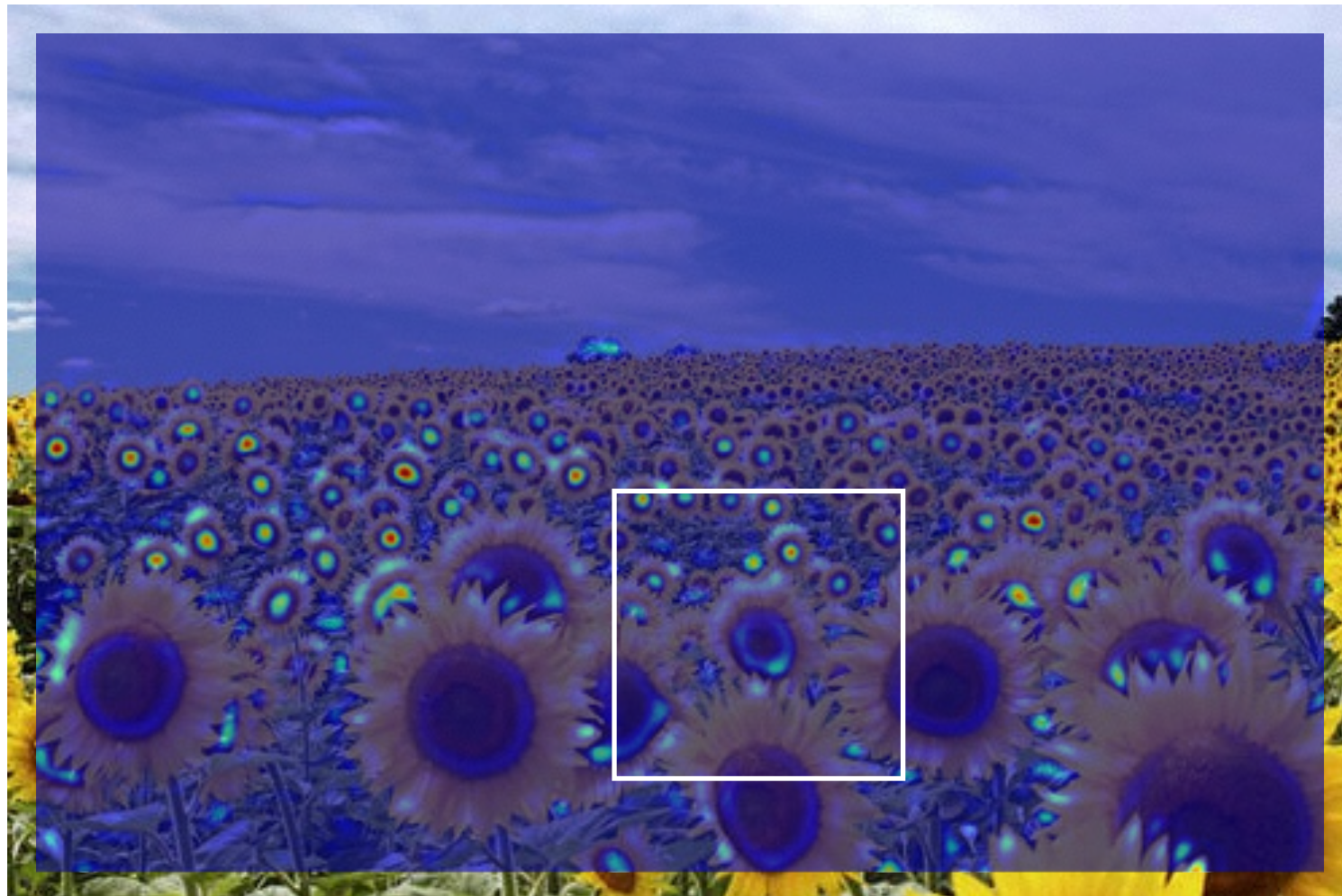
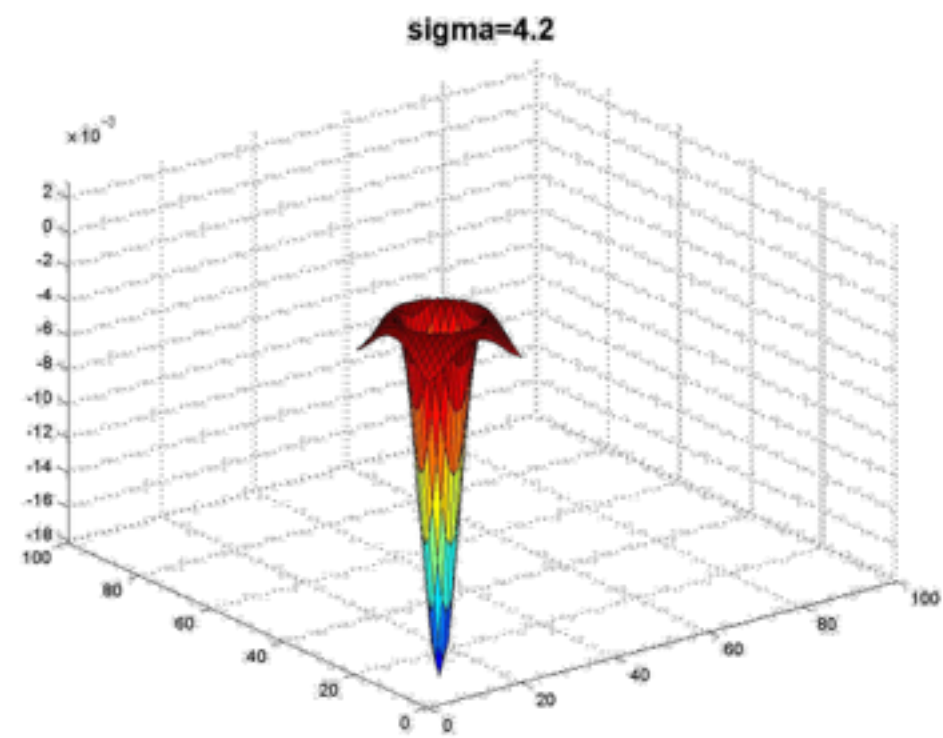
Full size

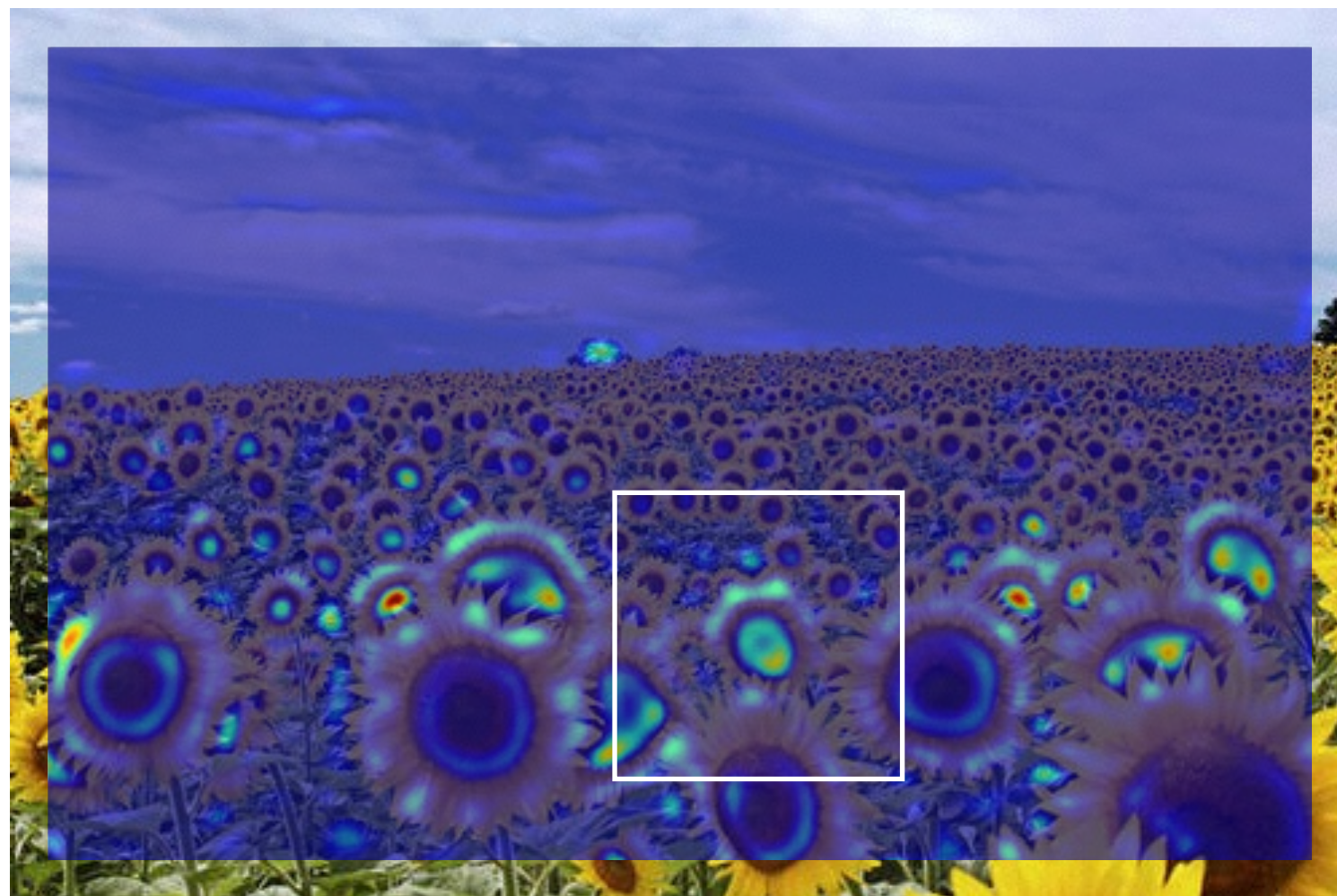
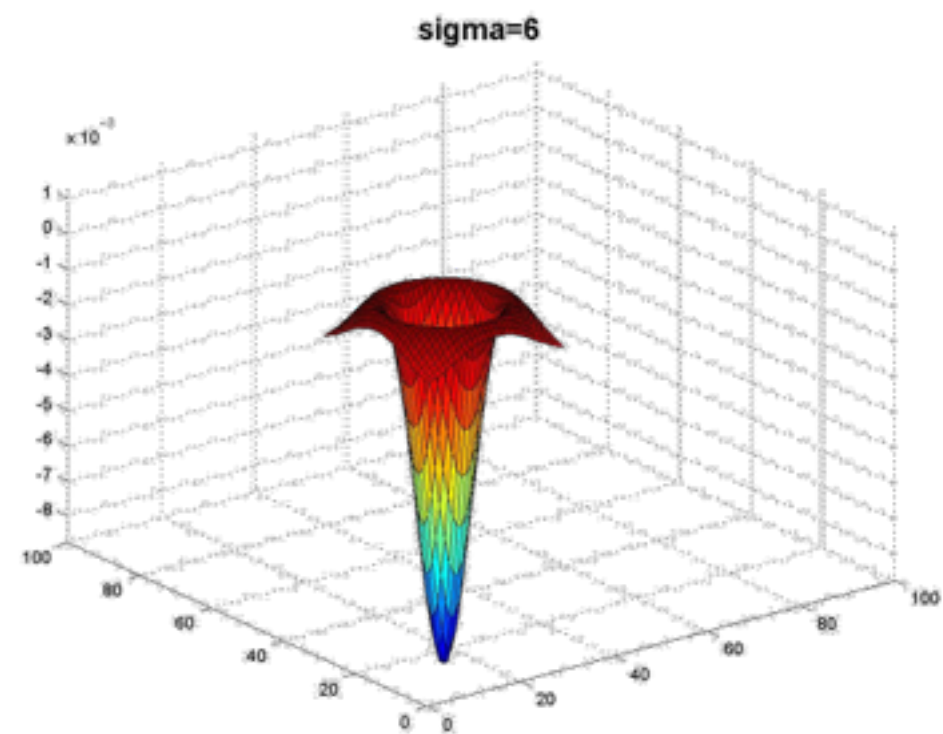


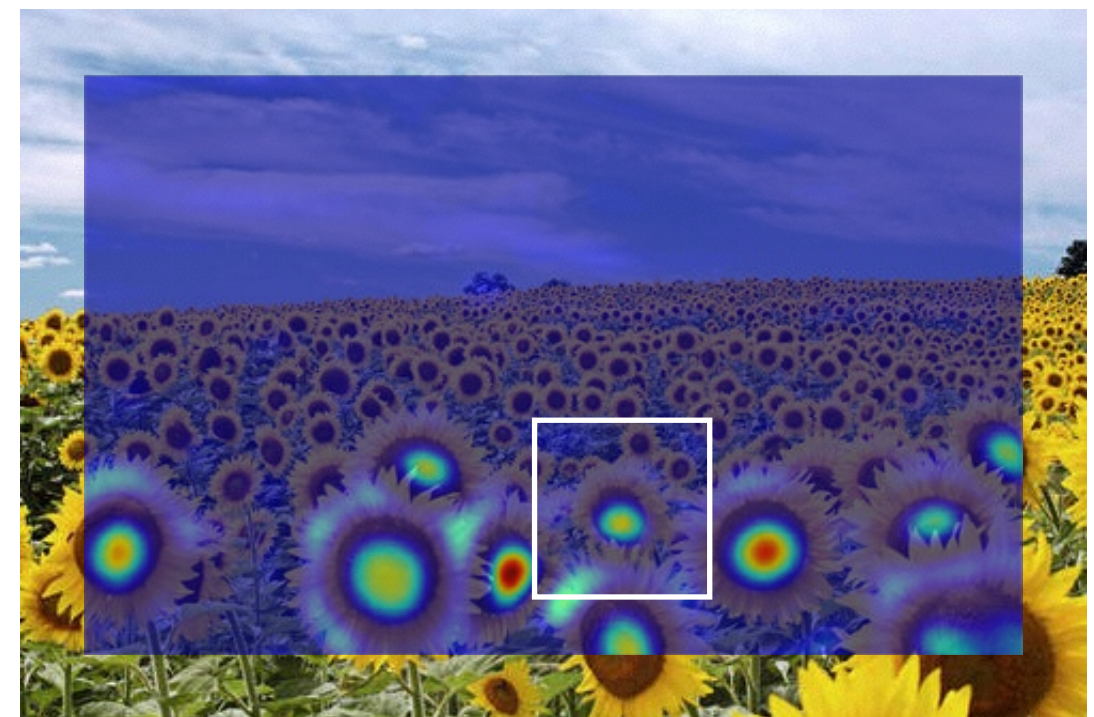
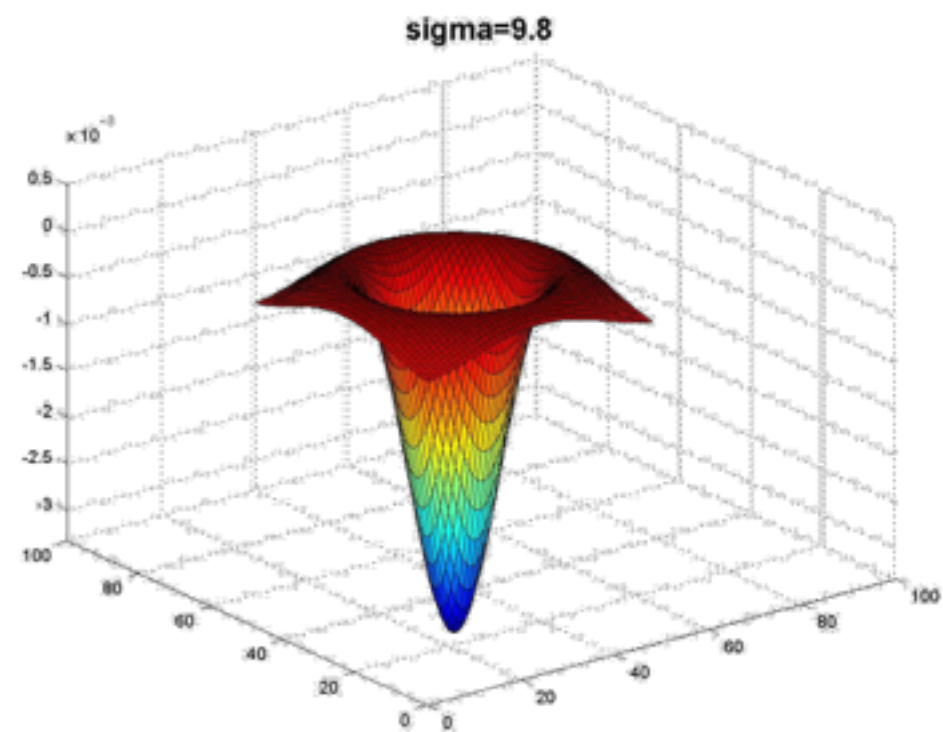
3/4 size

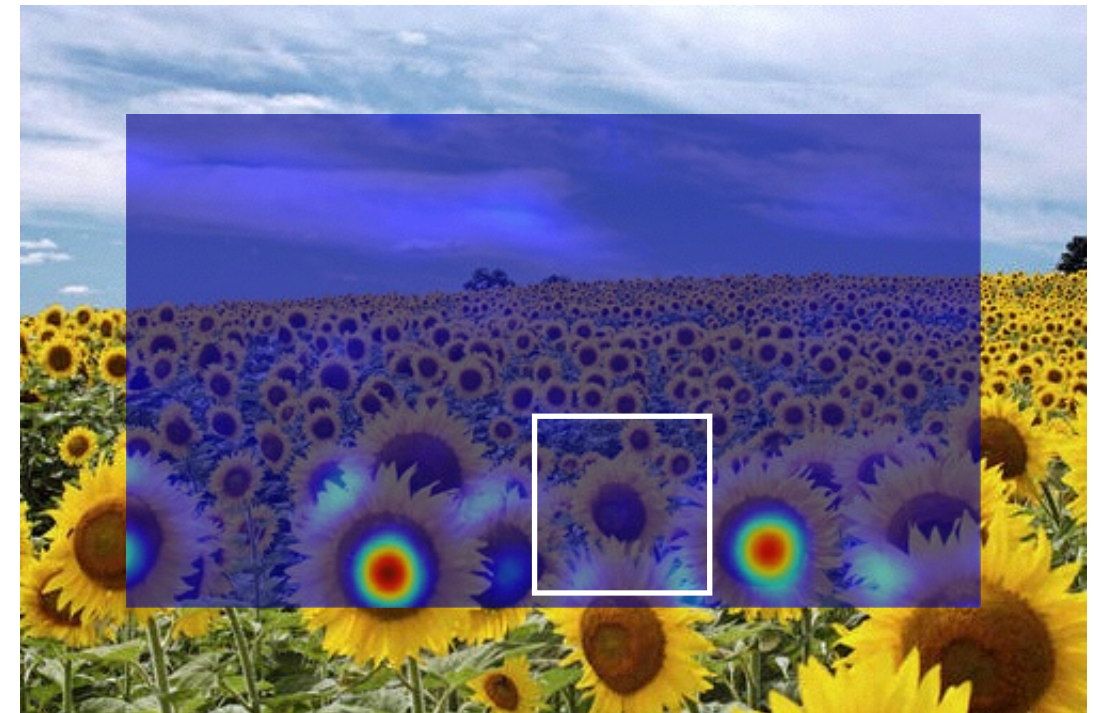
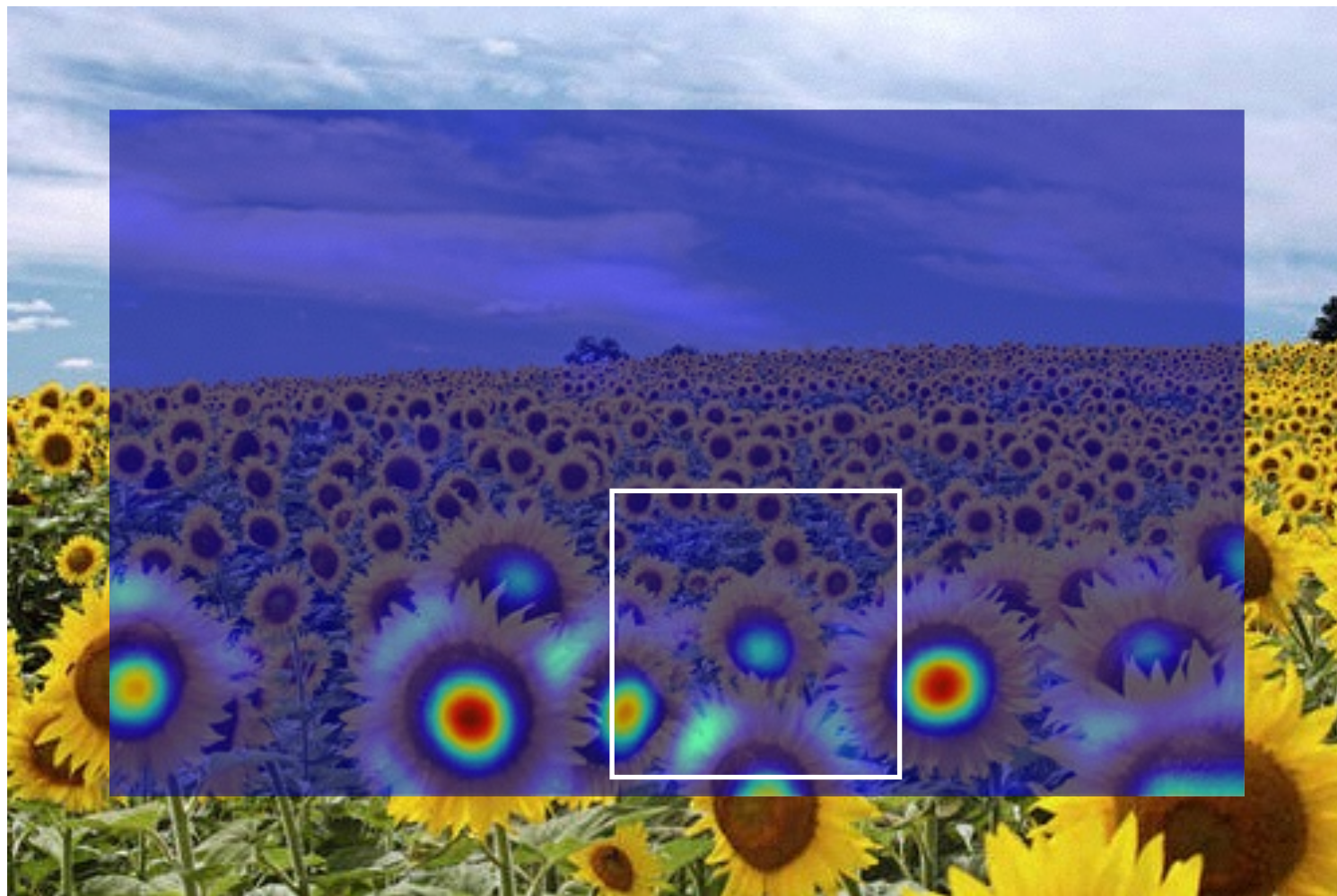
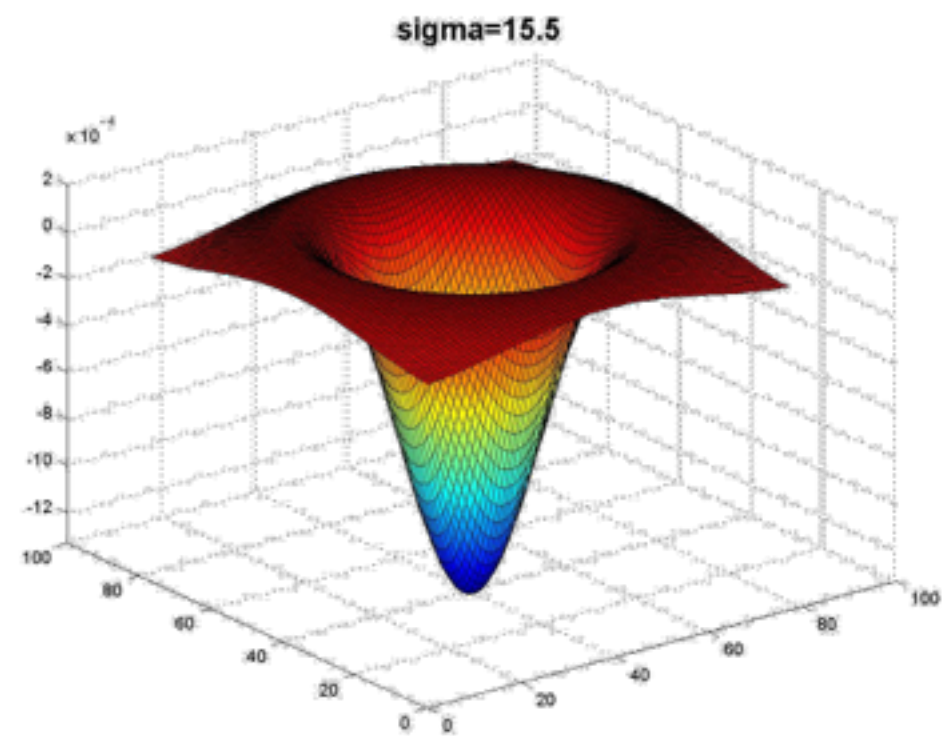


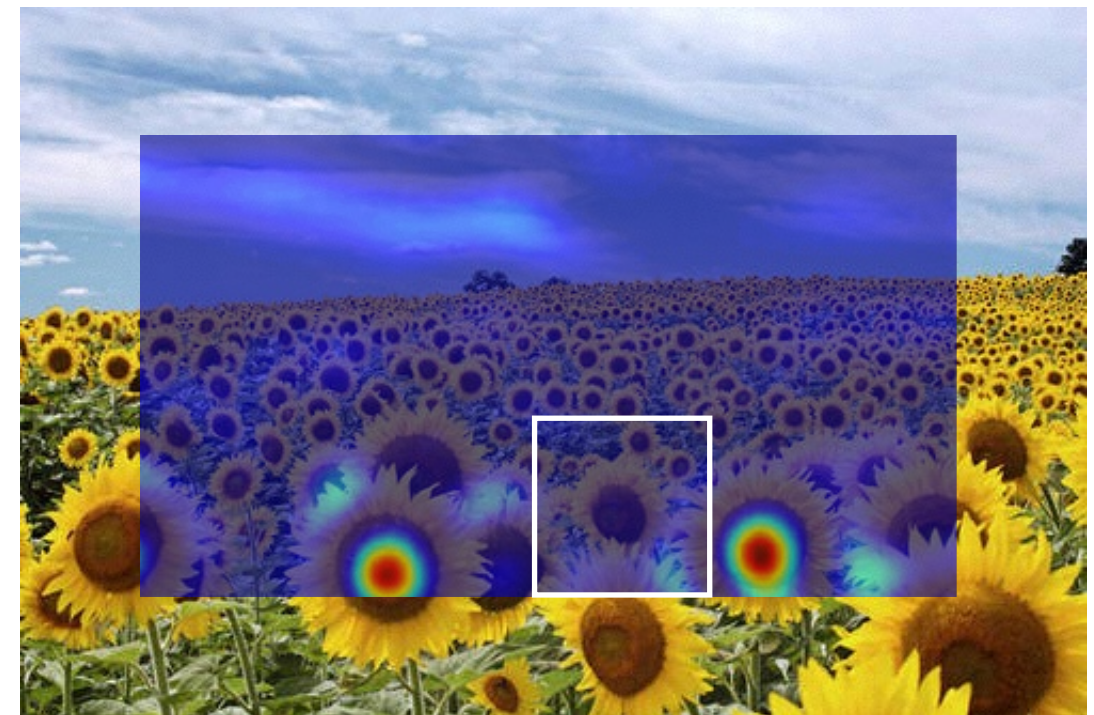
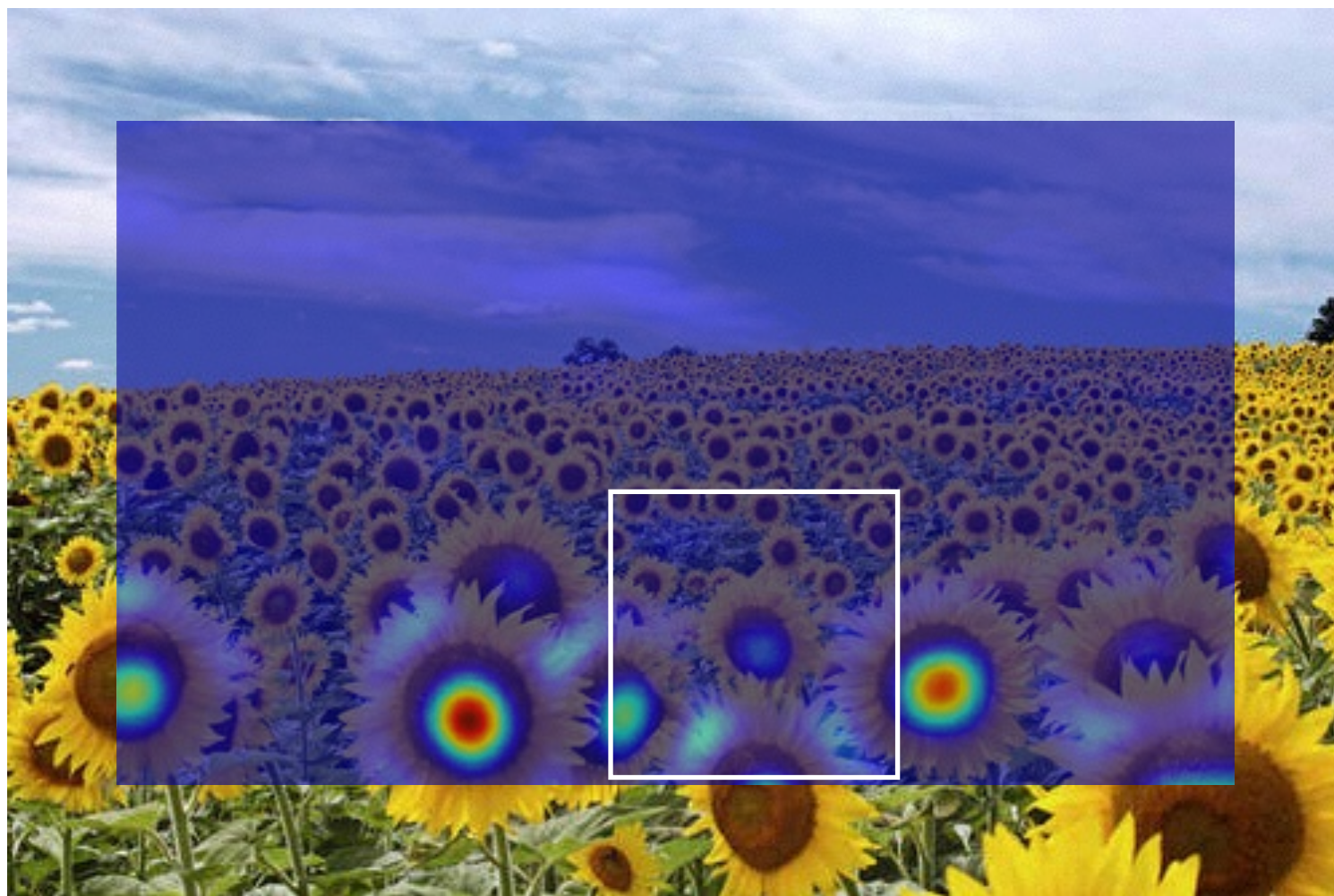
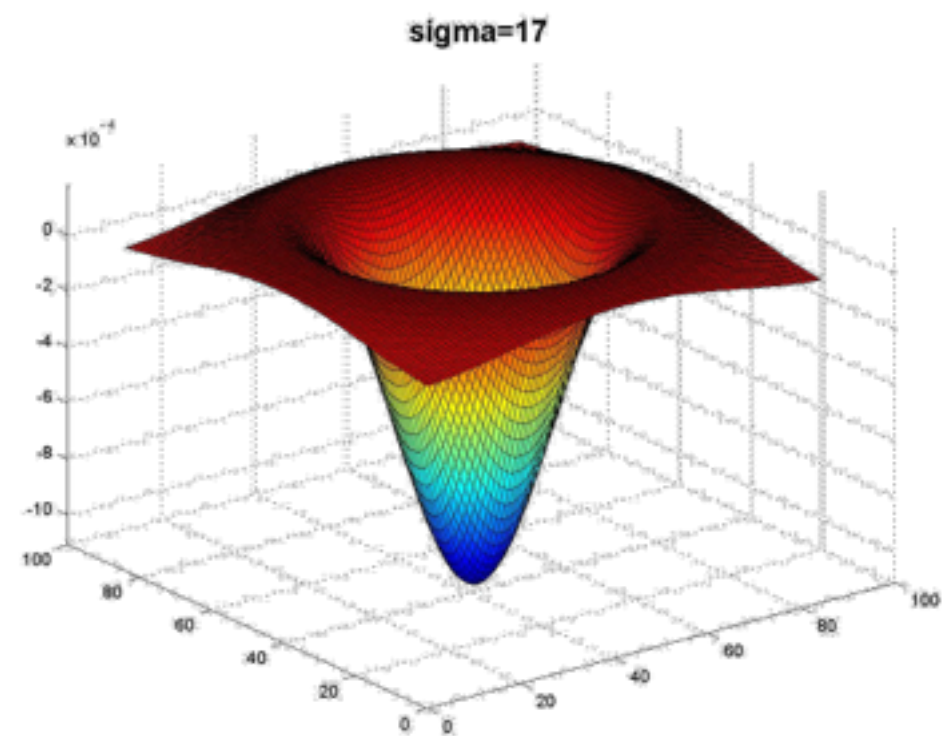












What happened when you applied different Laplacian filters?

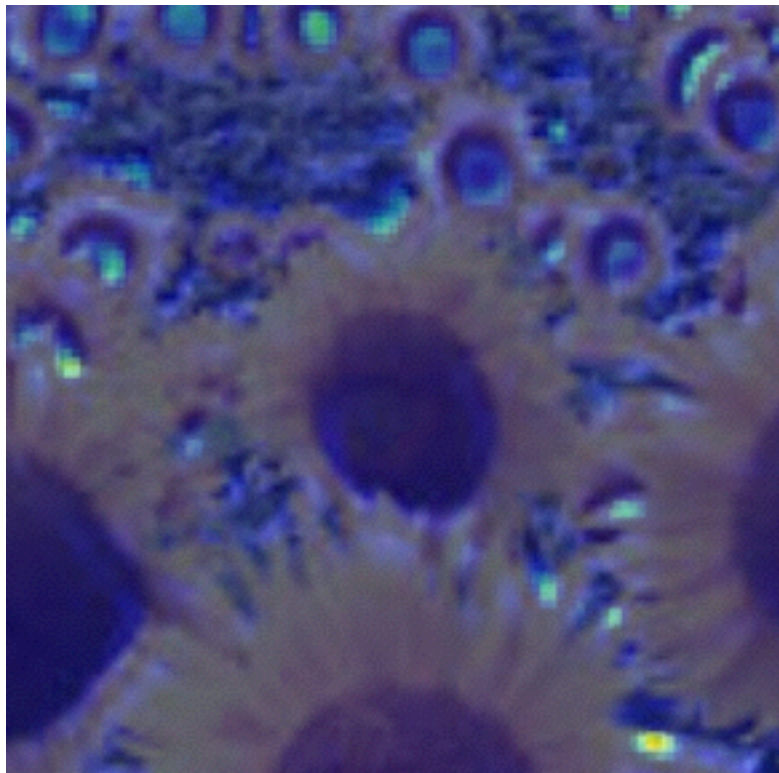
Full size



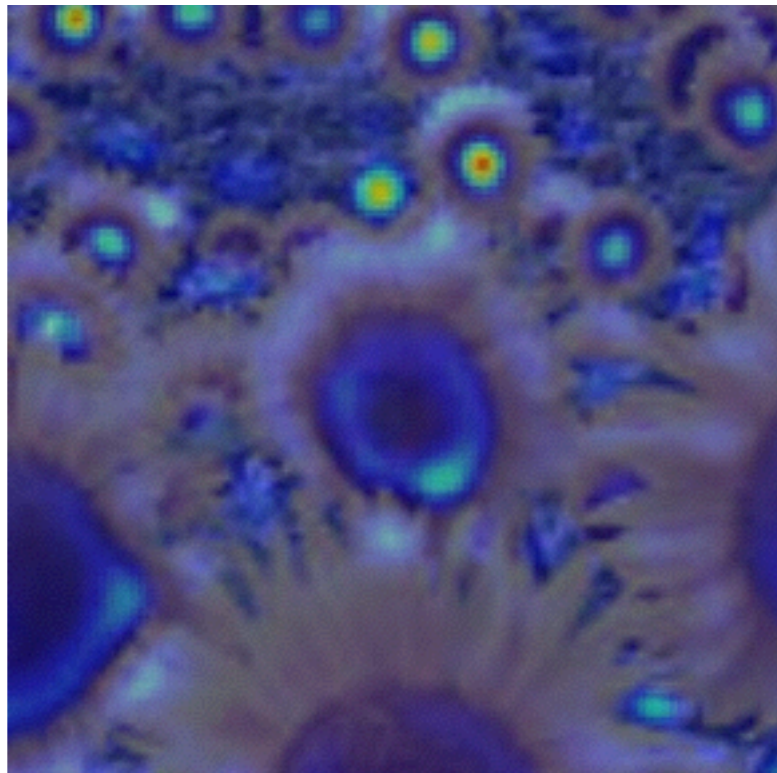
3/4 size



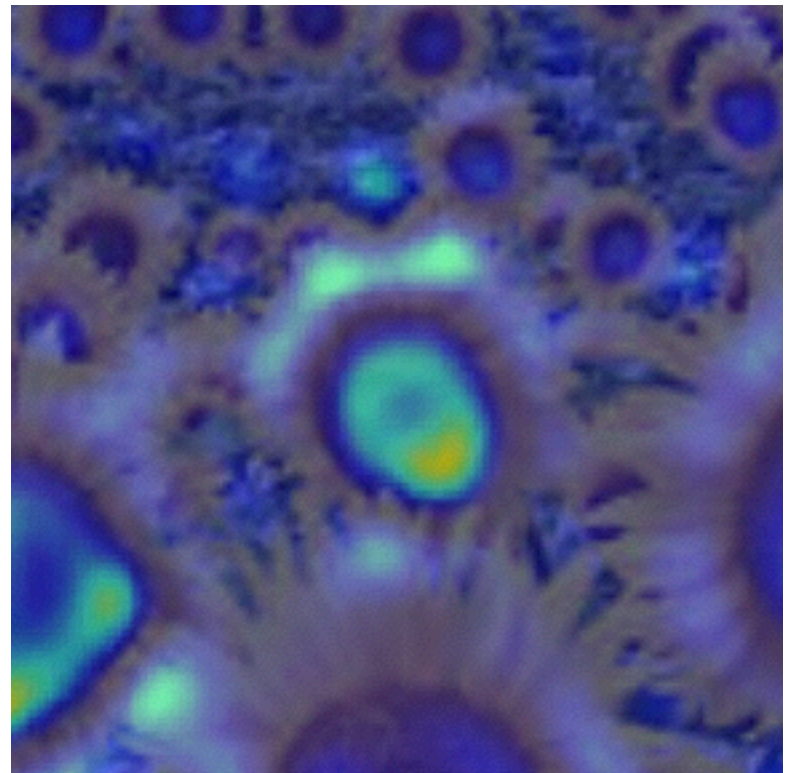
2.1



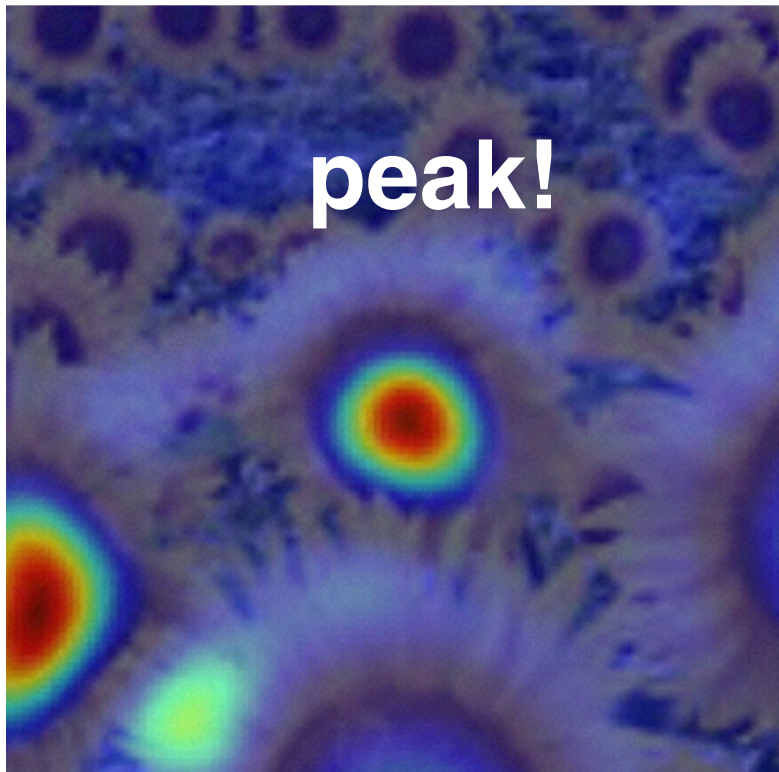
4.2



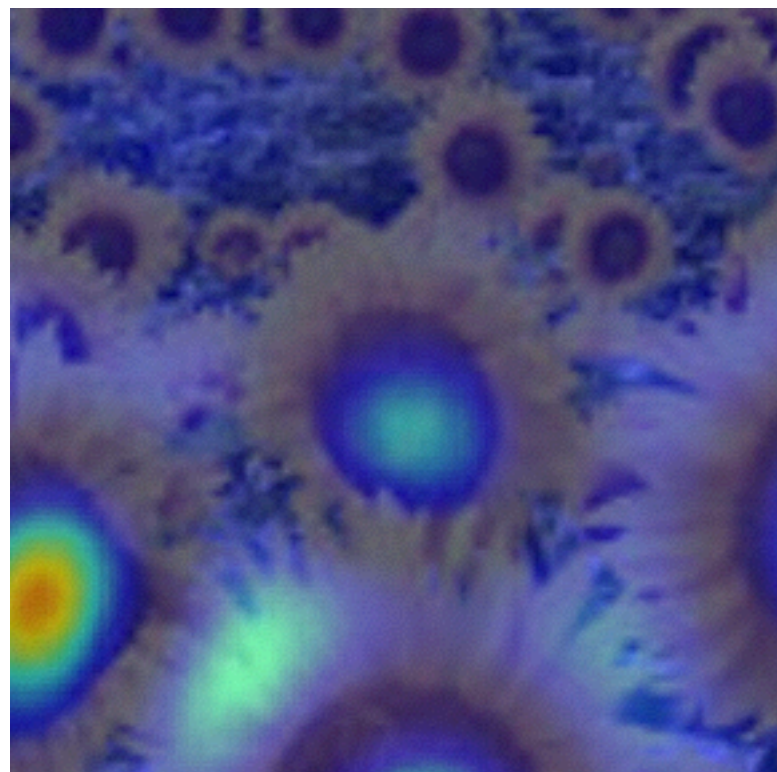
6.0



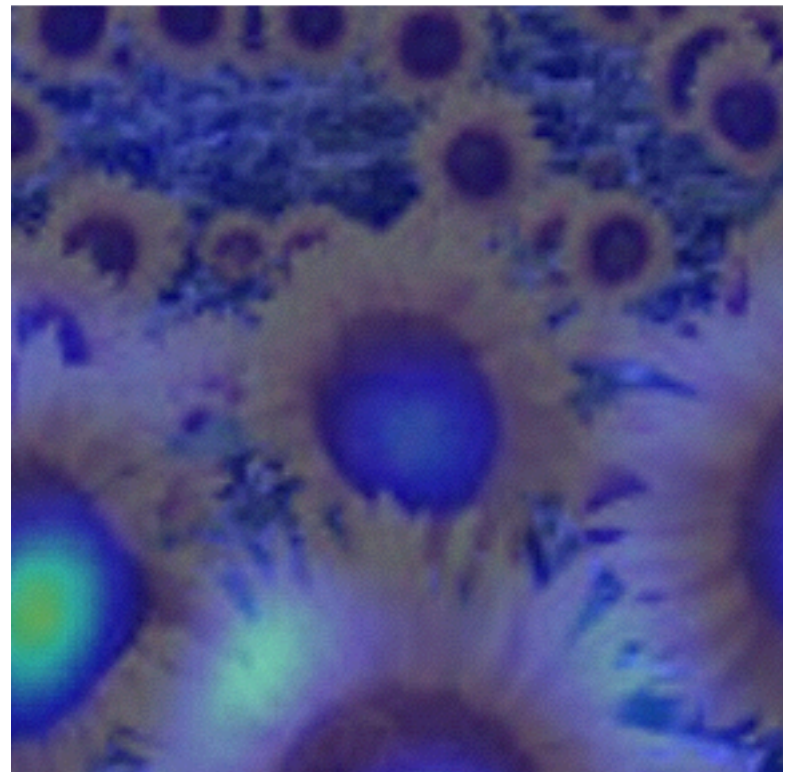
9.8



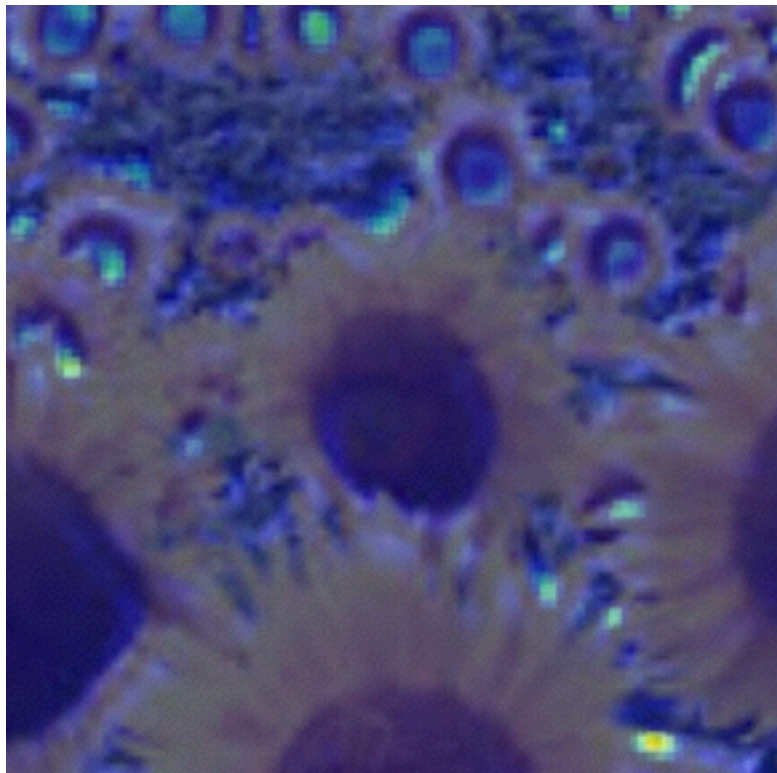
15.5



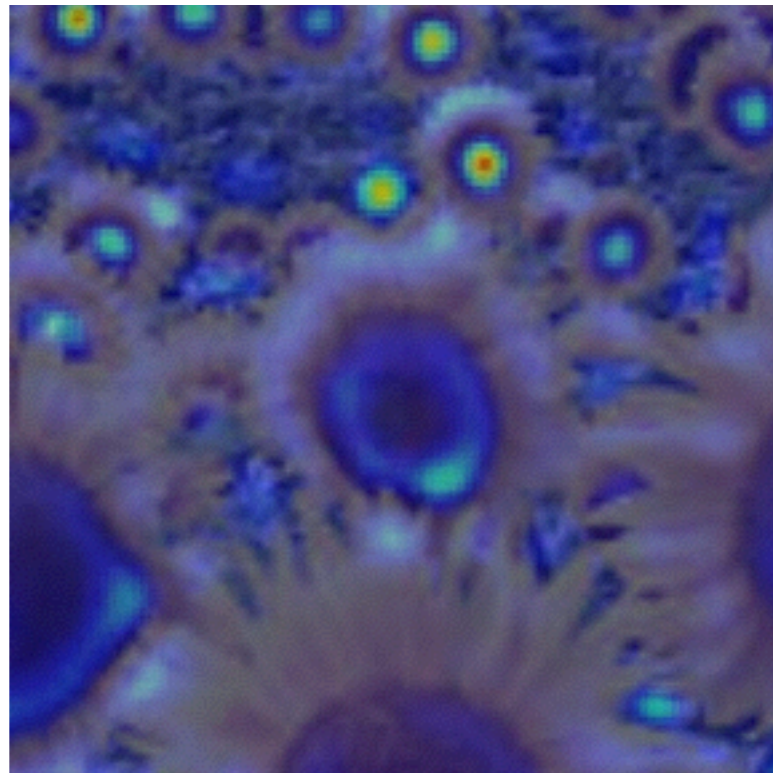
17.0



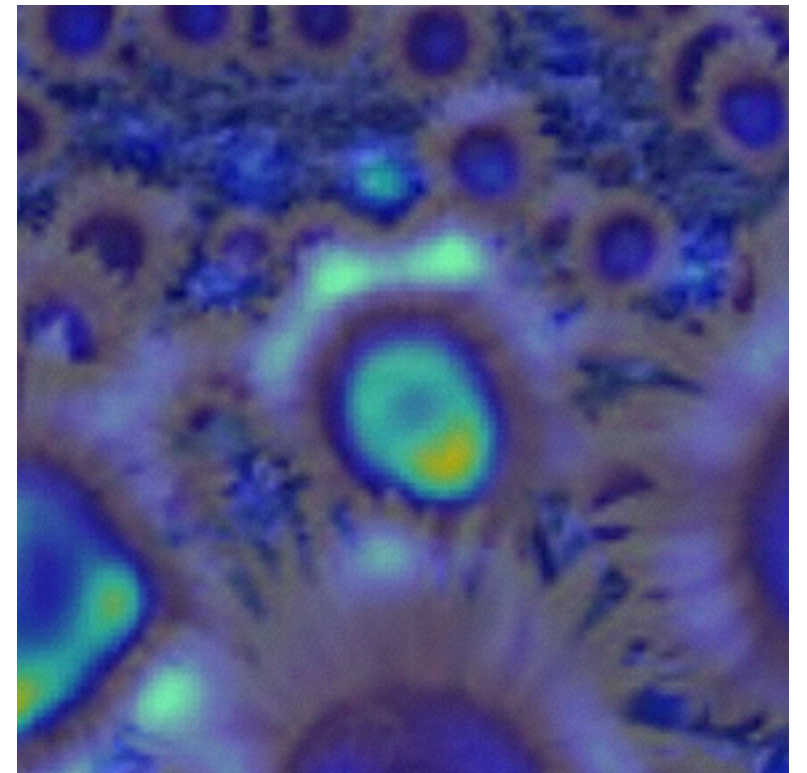
2.1



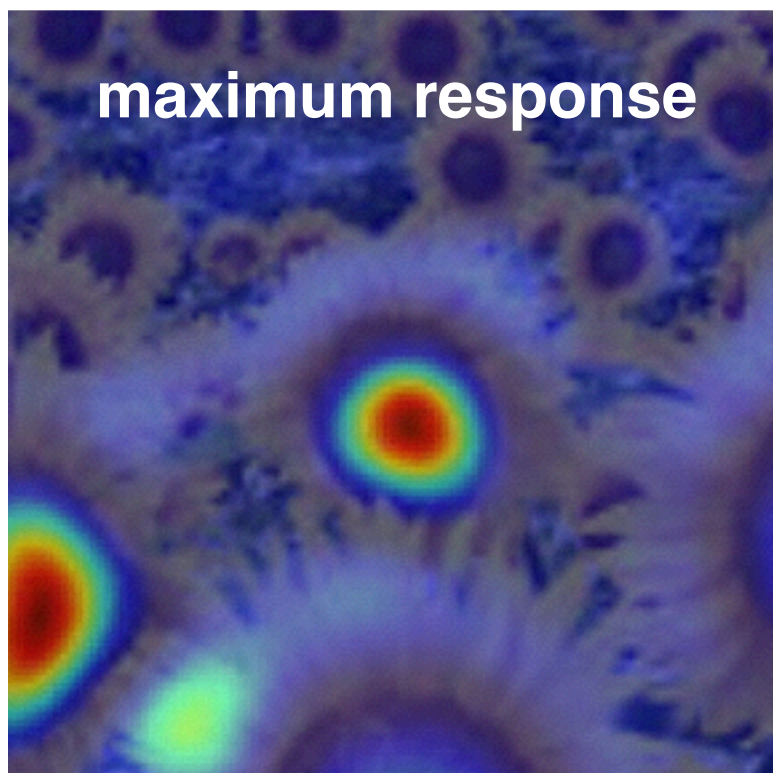
4.2



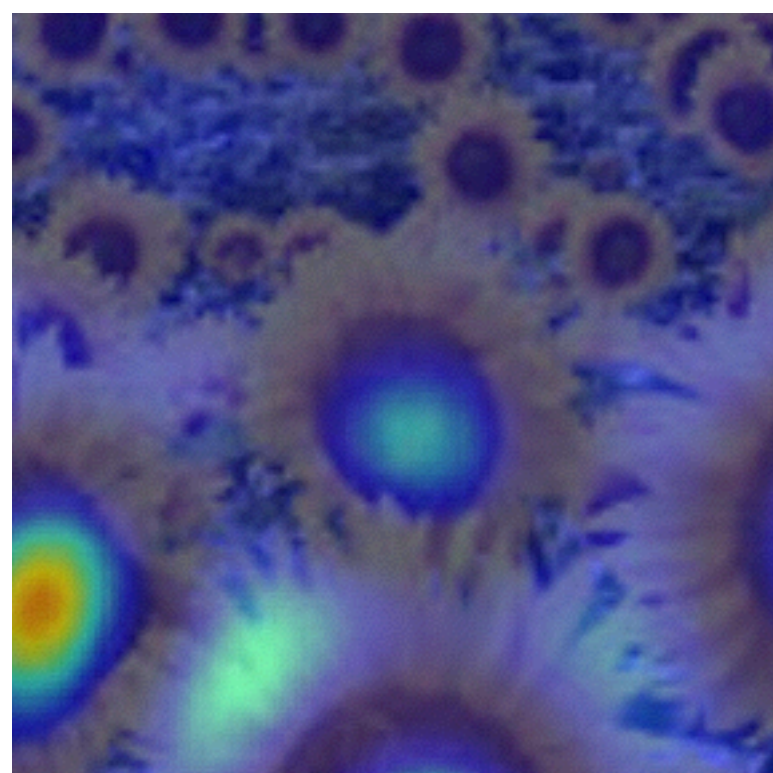
6.0



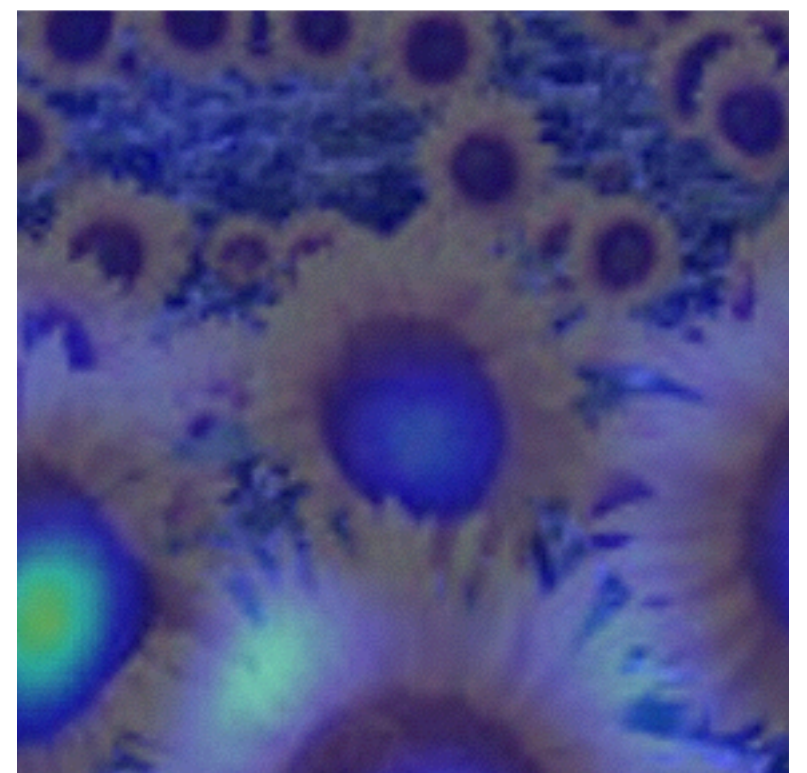
9.8



15.5

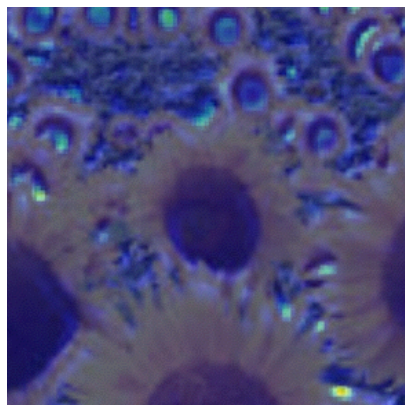


17.0

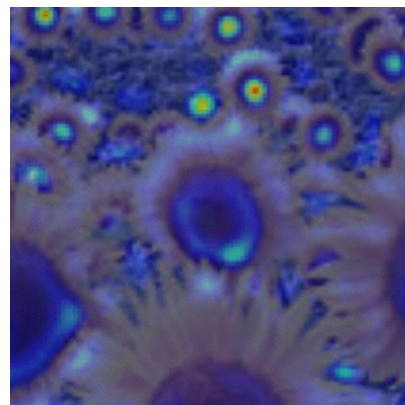


optimal scale

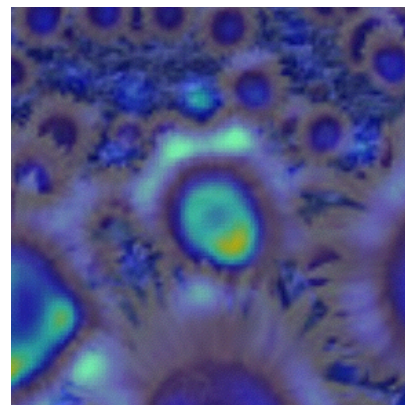
2.1



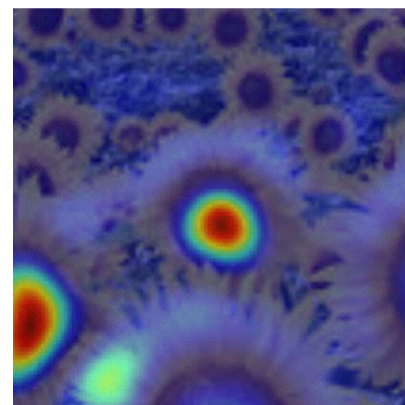
4.2



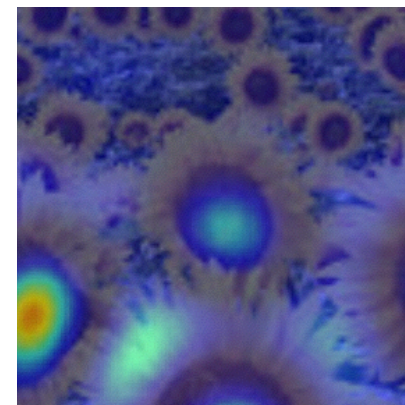
6.0



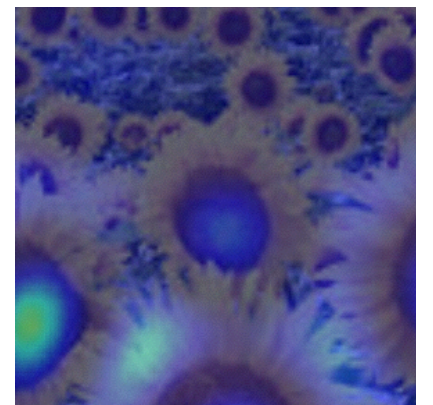
9.8



15.5

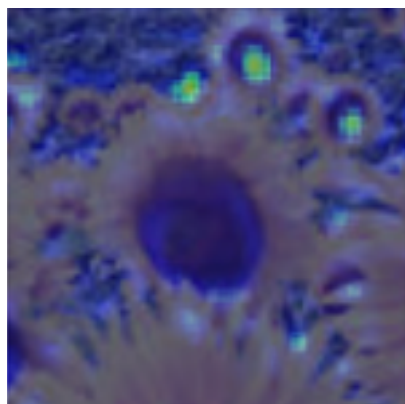


17.0

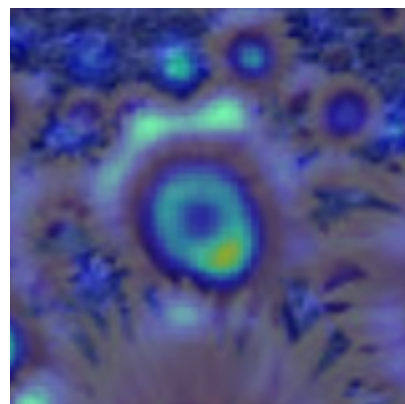


Full size image

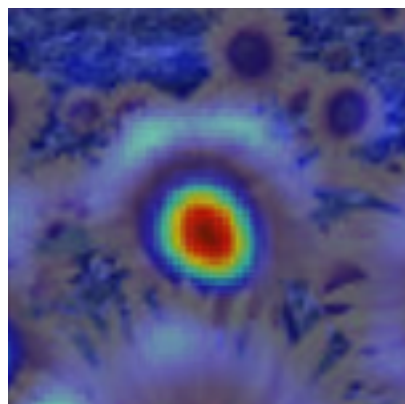
2.1



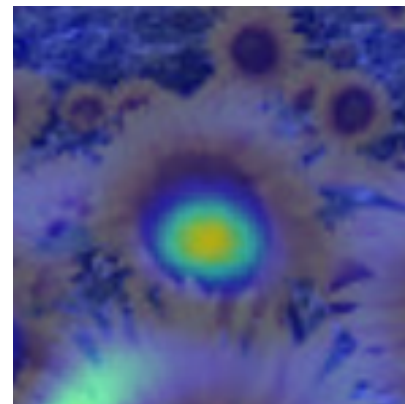
4.2



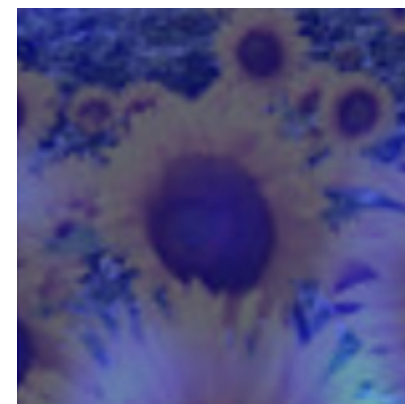
6.0



9.8



15.5



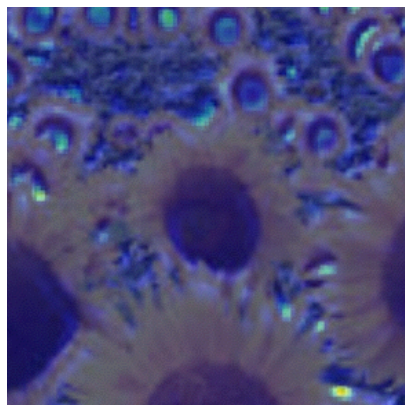
17.0



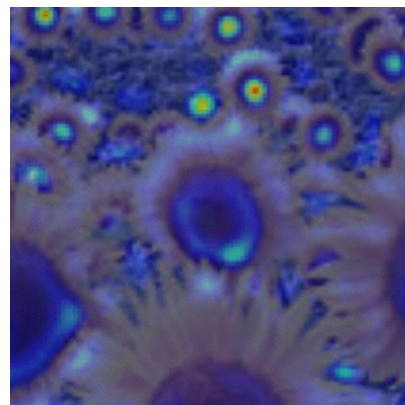
3/4 size image

optimal scale

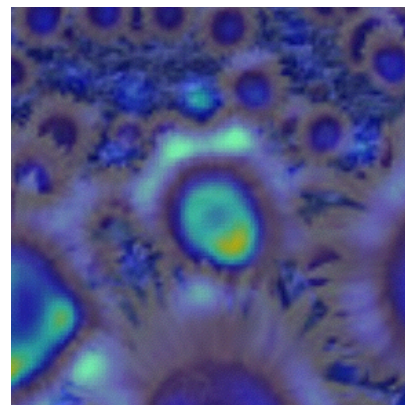
2.1



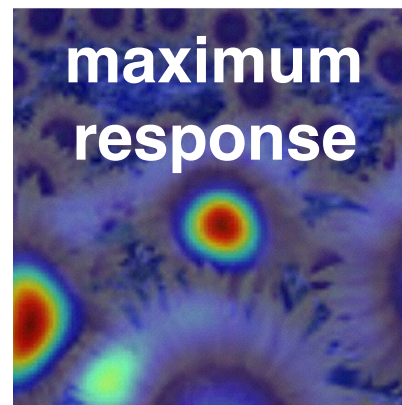
4.2



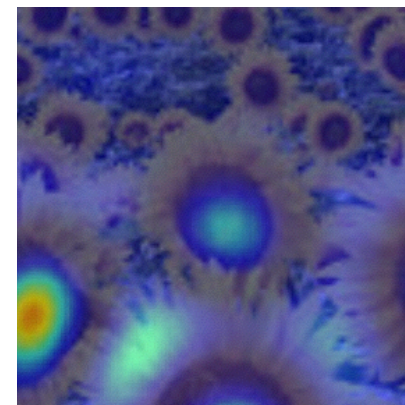
6.0



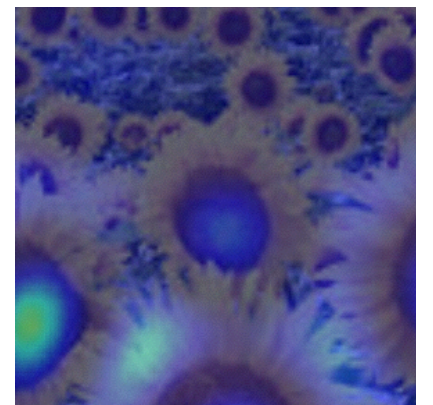
9.8



15.5

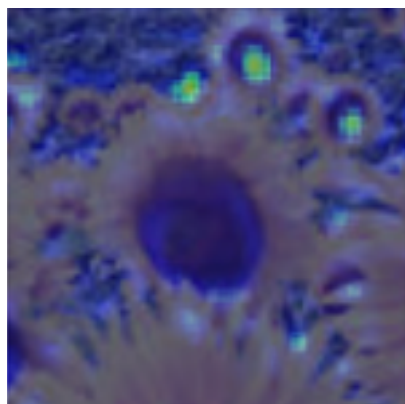


17.0

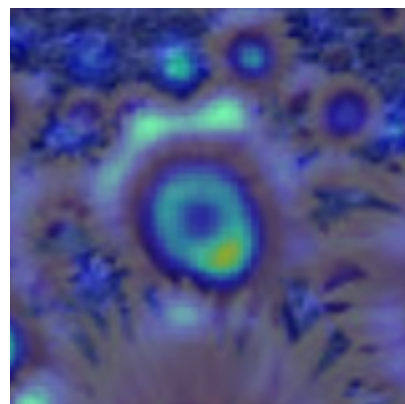


Full size image

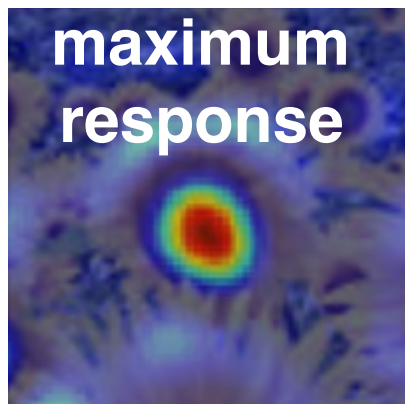
2.1



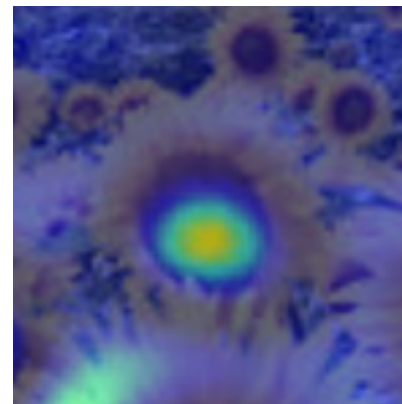
4.2



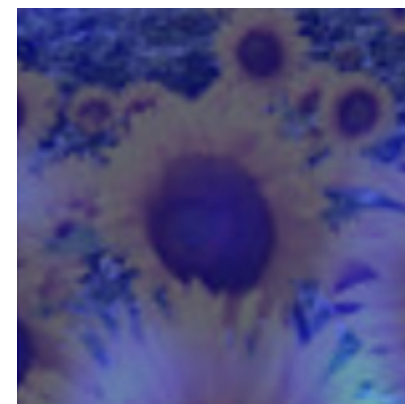
6.0



9.8



15.5

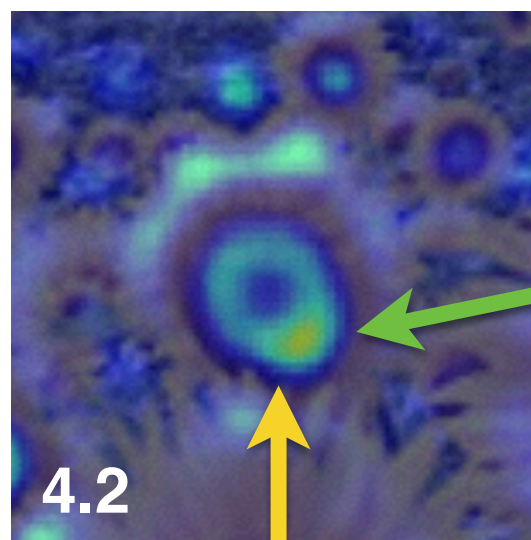


17.0

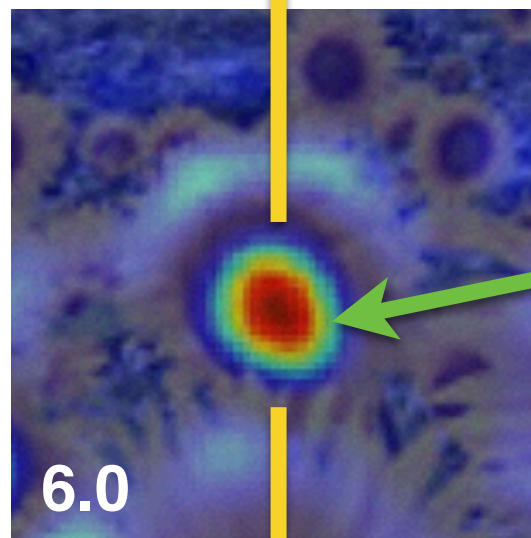


3/4 size image

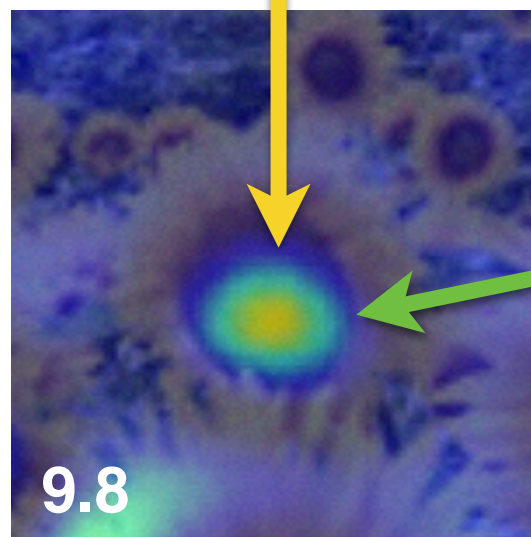
cross-scale maximum



local maximum



local maximum



local maximum

implementation

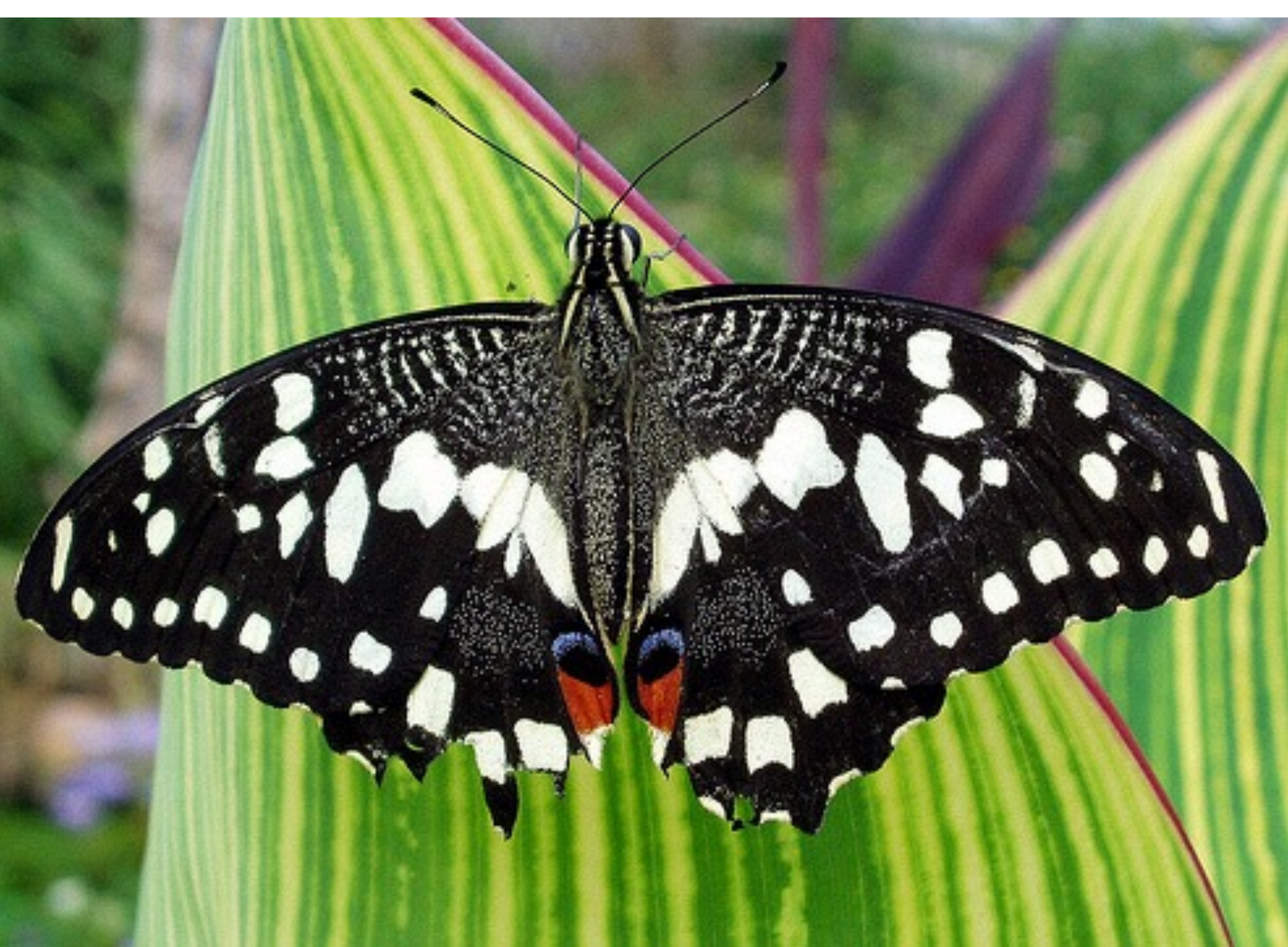
For each level of the Gaussian pyramid

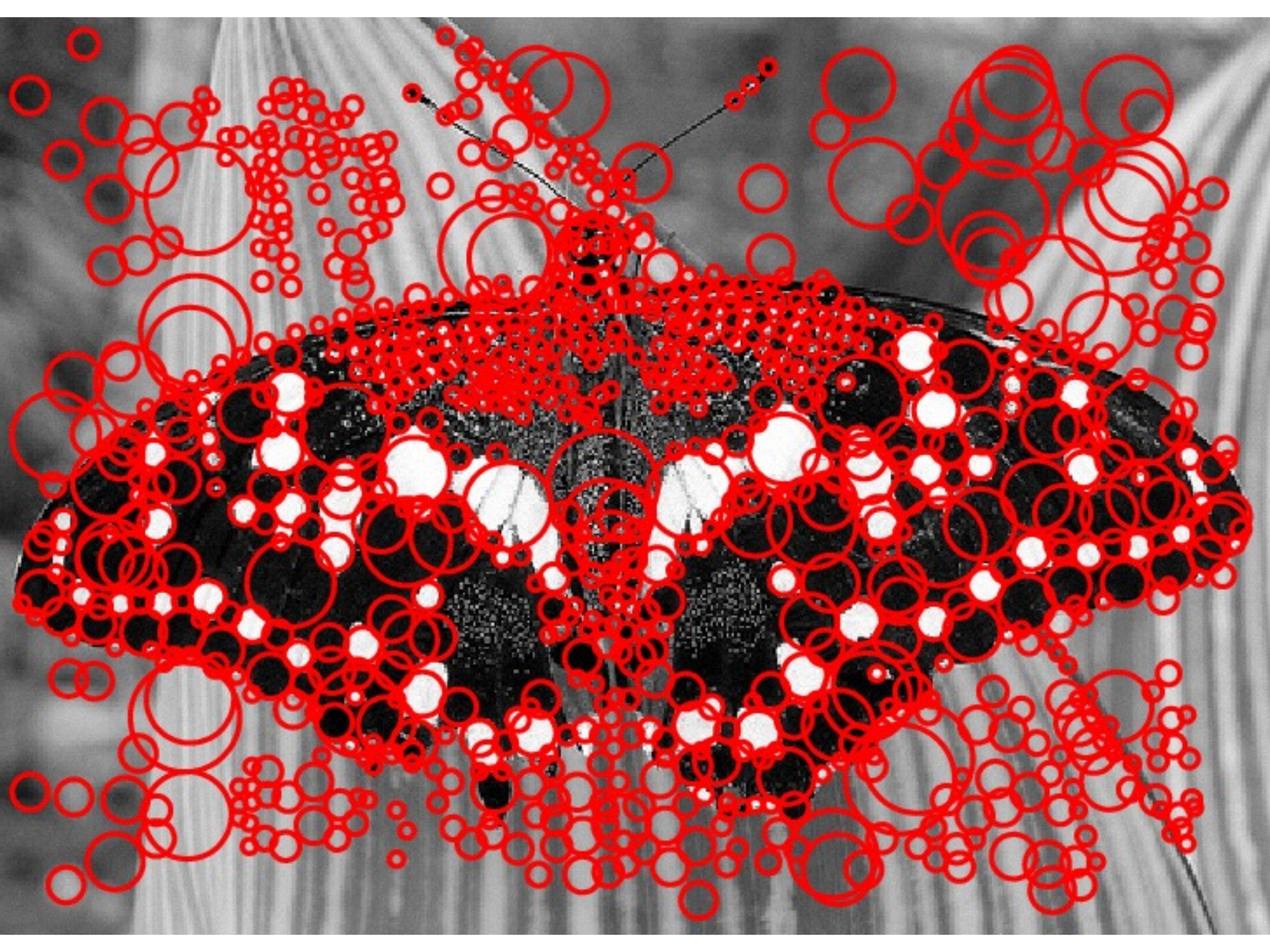
 compute feature response (e.g. Harris, Laplacian)

For each level of the Gaussian pyramid

 if local maximum and cross-scale

save scale and location of feature (x, y, s)





*We have **detected** ‘corners’ but what is this useful for?*

*We have **detected** ‘corners’ but what is this useful for?*

usually need to **match** points

*We have **detected** ‘corners’ but what is this useful for?*

usually need to **match** points

so we will need **descriptors**