Patches

Matthew McAvoy – in Partnership with Blur-de-blur team

2020.12.10

Classify blurred

$$\begin{bmatrix} 0 & 1 & 0 \\ 1 & -4 & 1 \\ 0 & 1 & 0 \end{bmatrix}$$

Figure 2: The Laplacian kernel.

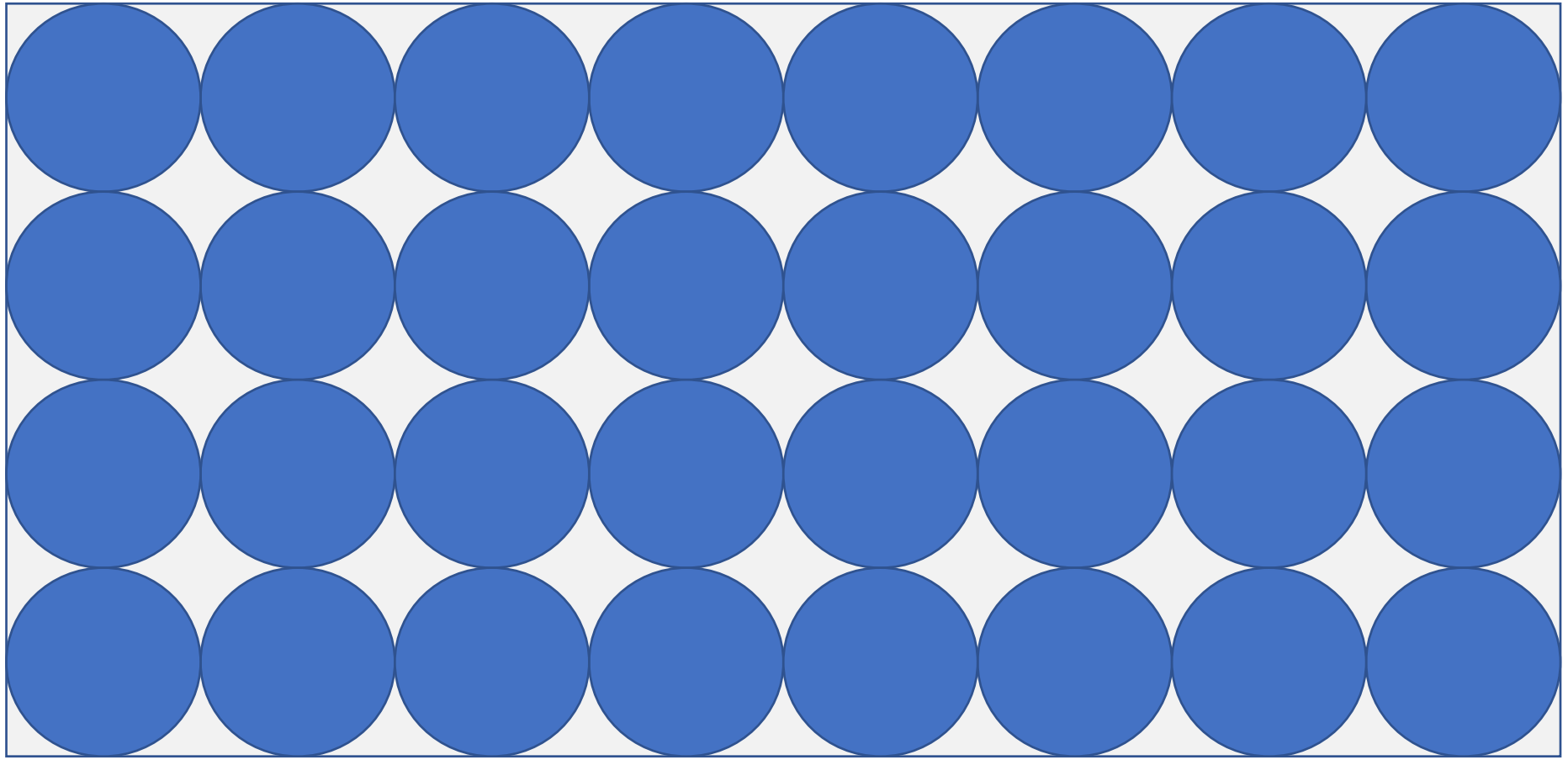
- A discrete laplacian is the sum of the second derivatives in a 3 by 3 convolution for each pixel
- Tells us about edges and motion in image
  - Many edges, higher variance -> classify not blurry
  - Fewer edges, lower variance -> classify blurry
- Hyperparameter on threshold constant – chose normalized variance
- Hand-drawn rectangles didn't fair very well; 5/9 correctly classified (errors in both positive and negative labels)

Source: https://www.pyimagesearch.com/2015/09/07/blur-detection-with-opencv/

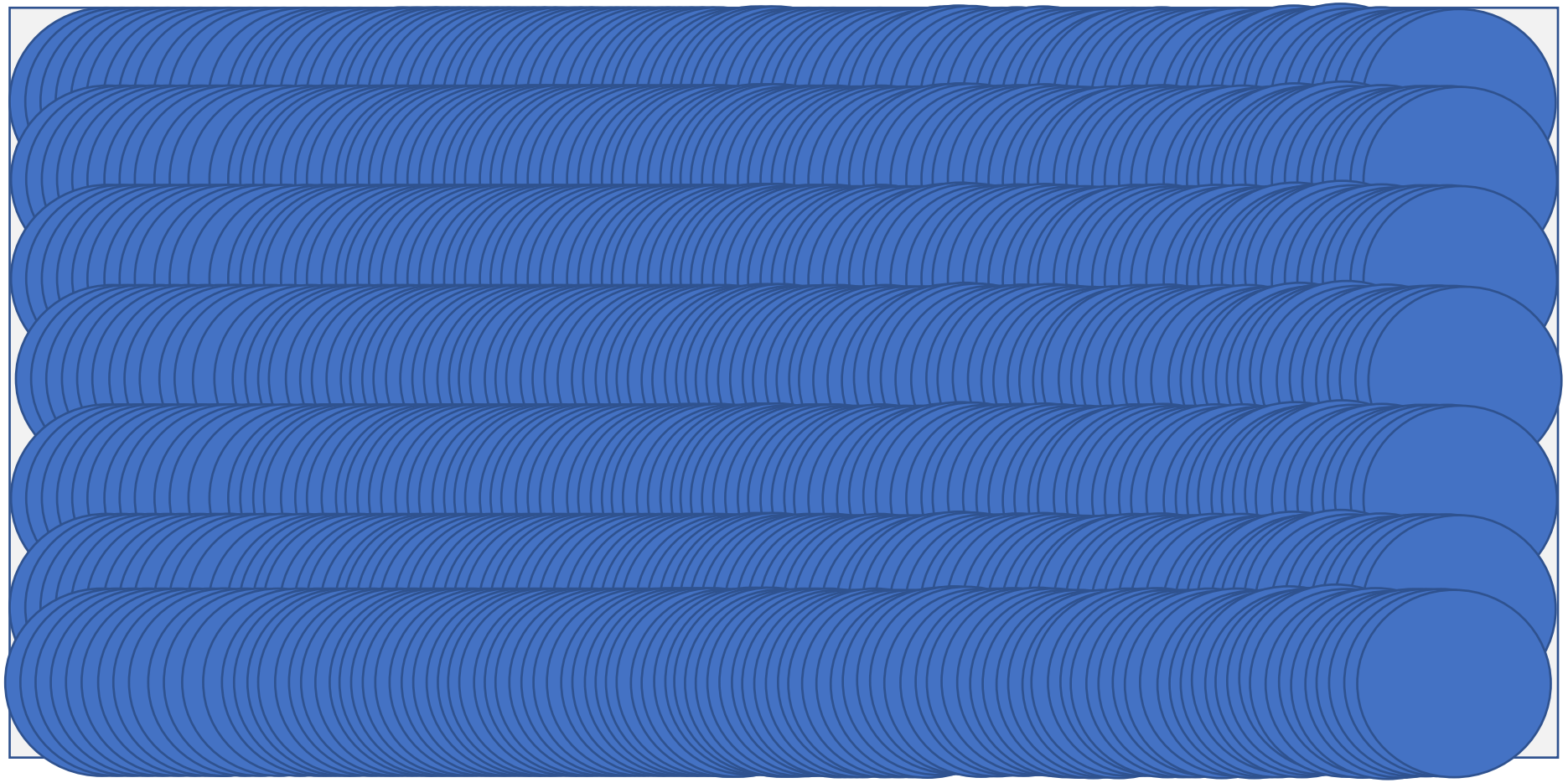
Examples

- Cover: 100 x 100
- Image: 4000 x 6000

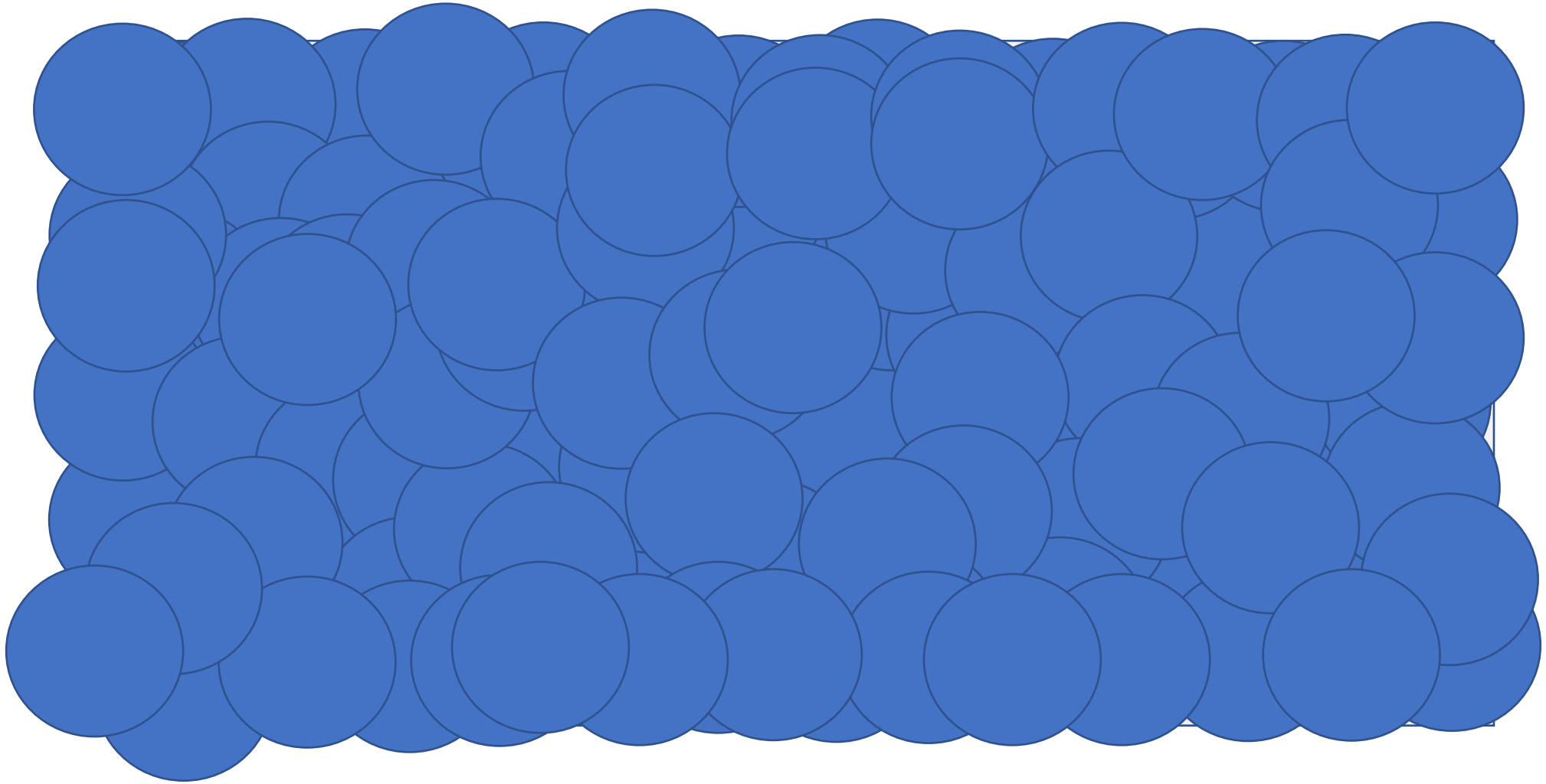
Lower bound - 2400



Upper bound – 23,010,000



Covering bound – ?

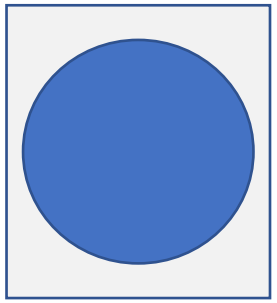


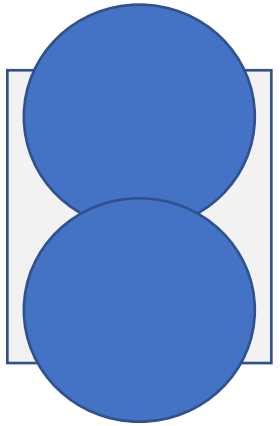
Covering Number Theorem

- A Covering number are the number of spherical balls of a fixed size required to completely cover a given space; overlaps allowed
- In Euclidean space $\mathbb{R}^m$. $\mathrm{K} \subset \mathbb{R}^m$ are the set of vectors whose length (norm) is at most k within a d - dimensional subspace, then:
- $N_r^{ext}(K) \leq \left(\frac{2k\sqrt{d}}{r}\right)^d$
- If we imagine the diagonal of our image is k, d is 2, and r is the diagonal of a fixed size patch, then we can calculate it!
- Covering number = 83,200

Post process

- Best way to cover?





Summary

- Blur region with fourier transform
- Classify blur using variance in laplacian

- Lower bound: 2,400
- Upper bound: 23,010,000
- Covering bound: 83,200