ECE 596

HW 2 Notes

K-means clustering

- Pixel-wise image segmentation in RGB color space.

K-means clustering

1.
Make a copy of your original image.

K-means clustering

1.
Make a copy of your original image.

Copying input image to a buffer image.

K-means clustering

1.
Make a copy of your original image.

2.
Initialize cluster centers.
double** centers

centers[i][j]:
,where i = 1 … number of clusters
j = 1 … 3 (R,G,B channels)

K-means clustering

1.
Make a copy of your original image.

2.
Initialize cluster centers.
double** centers

centers[i][j]:
,where i = 1 … number of clusters
j = 1 … 3 (R,G,B channels)

Task 1 : random seeds
Task 2: pixel seeds
Task 3: histogram seeds

K-means clustering

1.
Make a copy of your original image.

2.
Initialize cluster centers.
double** centers

3.
Find closest cluster for each pixel.
|Rp-Rc|+|Gp-Gc|+|Bp-Bc|

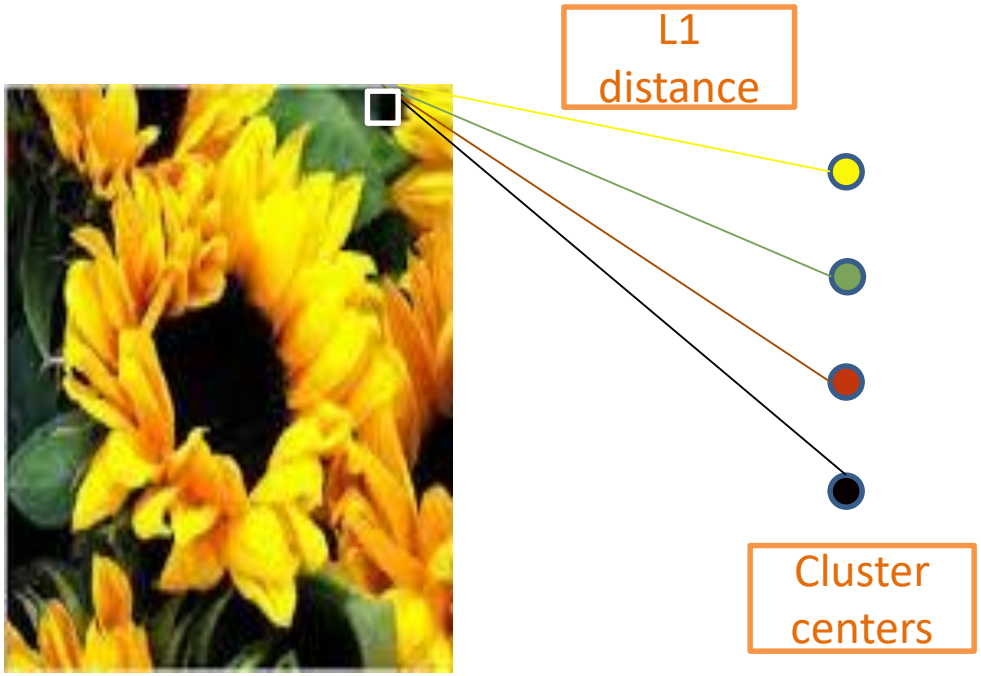
Cluster centers

K-means clustering

1.
Make a copy of your original image.

2.
Initialize cluster centers.
double** centers

3.
Find closest cluster for each pixel.
|Rp-Rc|+|Gp-Gc|+|Bp-Bc|

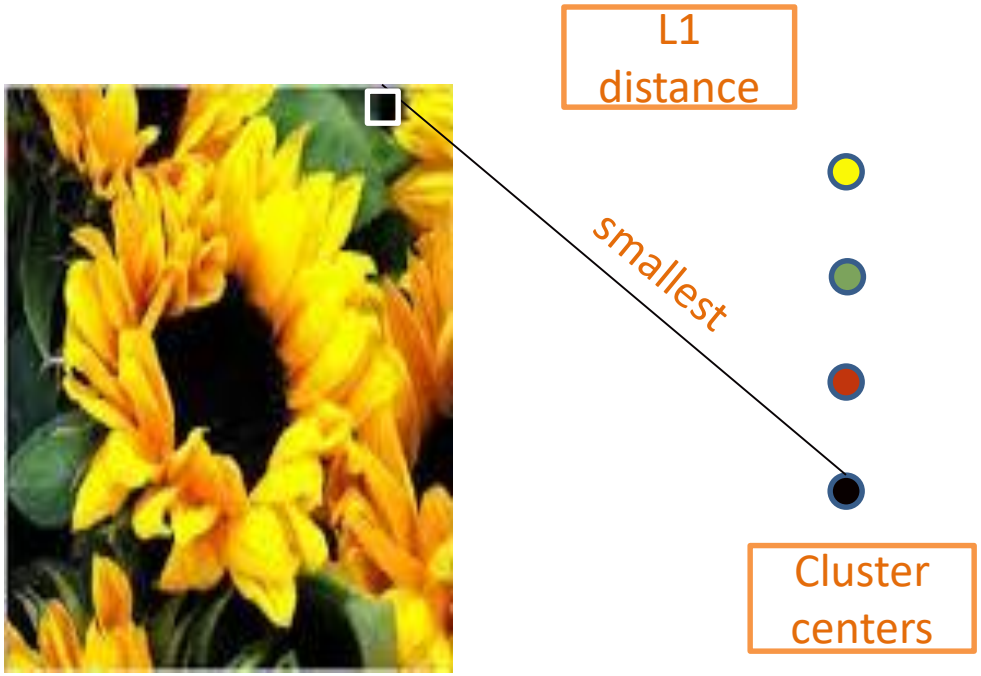


K-means clustering

1.
Make a copy of your original image.

2.
Initialize cluster centers.
double** centers

3.
Find closest cluster for each pixel.
|Rp-Rc|+|Gp-Gc|+|Bp-Bc|

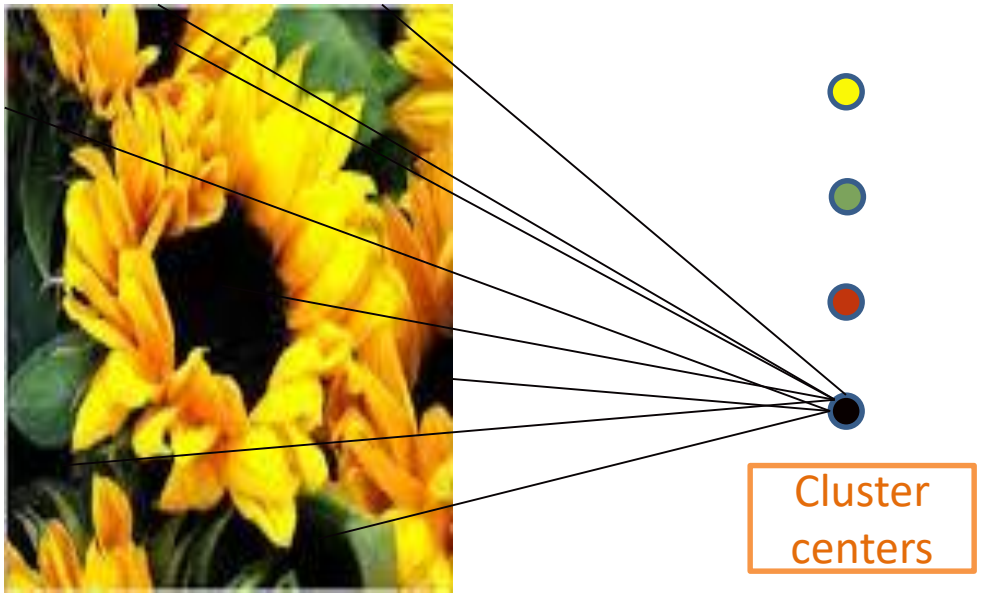


K-means clustering

1.
Make a copy of your original image.

2.
Initialize cluster centers.
double** centers

3.
Find closest cluster for each pixel.
|Rp-Rc|+|Gp-Gc|+|Bp-Bc|

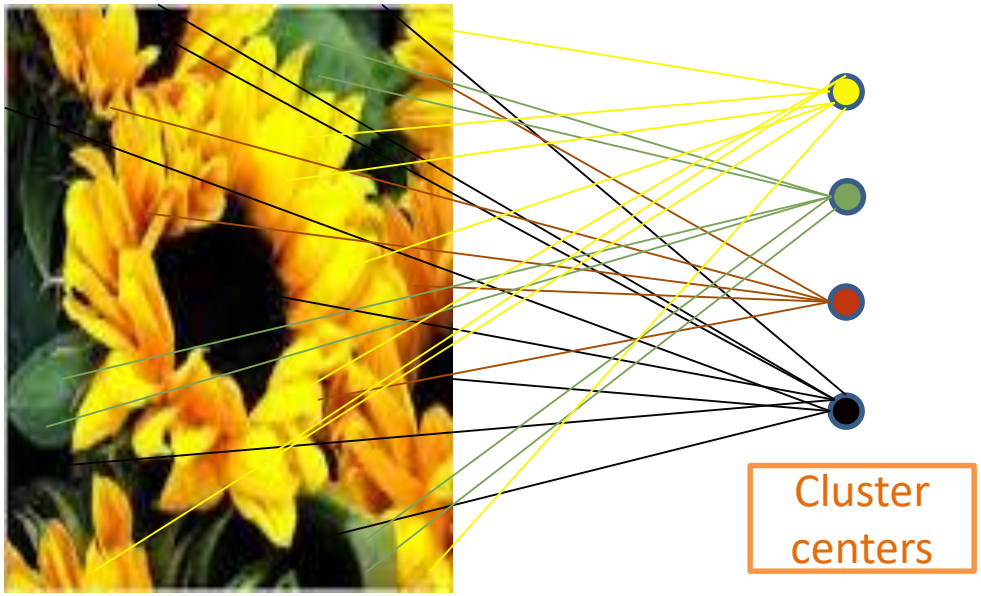


K-means clustering

1.
Make a copy of your original image.

2.
Initialize cluster centers.
double** centers

3.
Find closest cluster for each pixel.
|Rp-Rc|+|Gp-Gc|+|Bp-Bc|

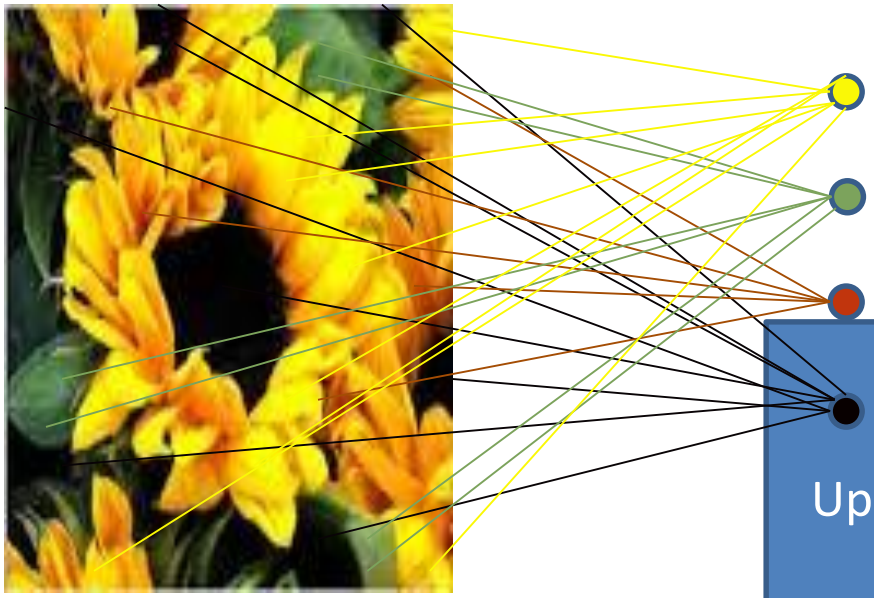


K-means clustering

1.
Make a copy of your original image.

2.
Initialize cluster centers.
double** centers

3.
Find closest cluster for each pixel.
|Rp-Rc|+|Gp-Gc|+|Bp-Bc|



4.
Update cluster centers.

K-means clustering

1.
Make a copy of your original image.

2.
Initialize cluster centers.
double** centers

3.
Find closest cluster for each pixel.
|Rp-Rc|+|Gp-Gc|+|Bp-Bc|

4.
Update cluster centers.

Take the mean and update cluster center color.

K-means clustering

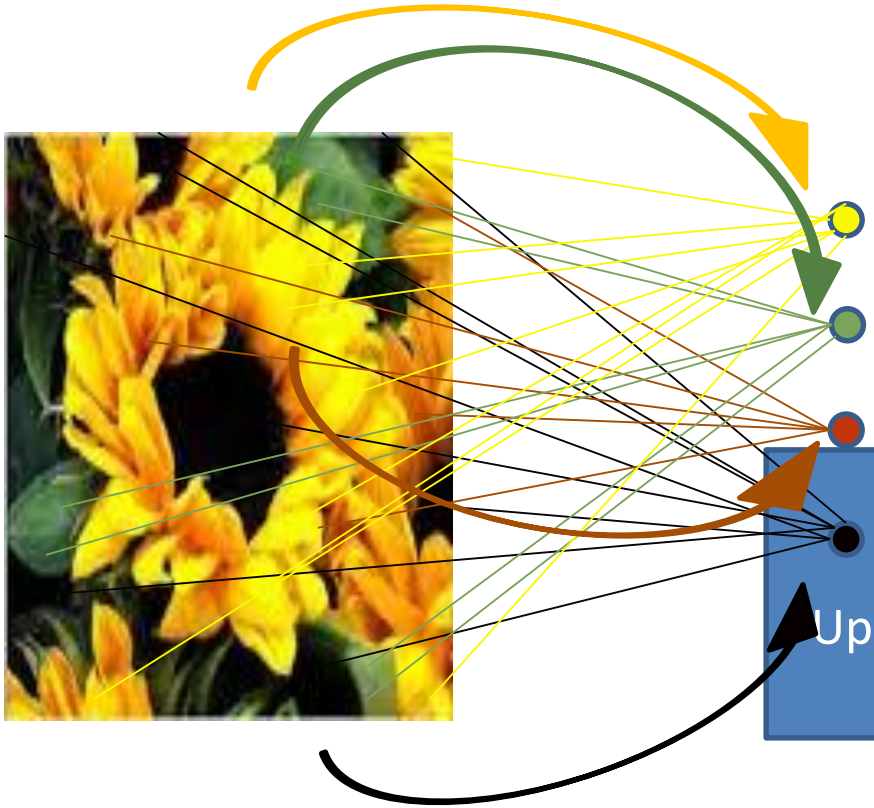
1.
Make a copy of your original image.

2.
Initialize cluster centers.
double** centers

3.
Find closest cluster for each pixel.
|Rp-Rc|+|Gp-Gc|+|Bp-Bc|

4.
Update cluster centers.

Do the same for all cluster centers.



K-means clustering

1.
Make a copy of your original image.

2.
Initialize cluster centers.
double** centers

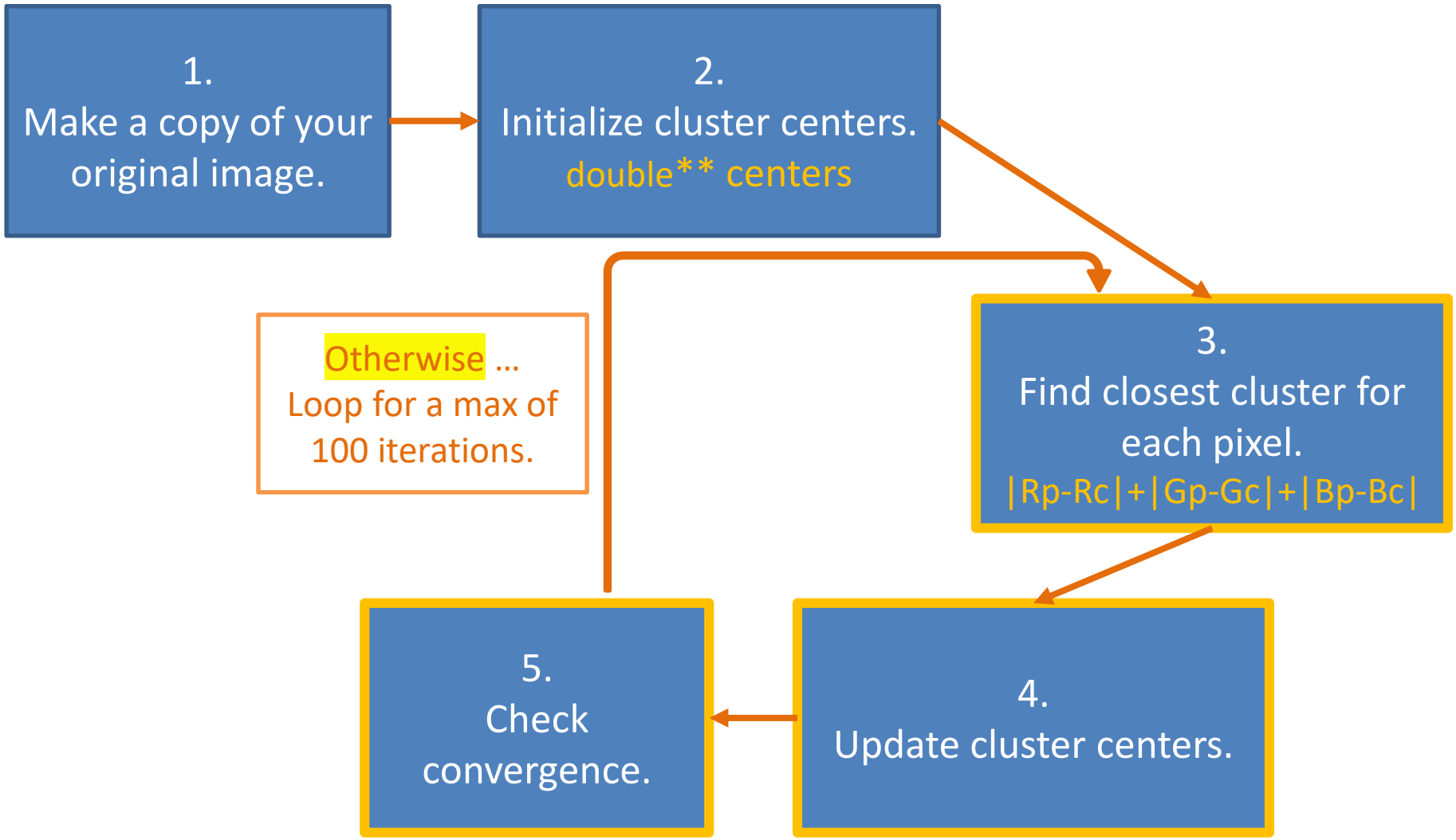
3.
Find closest cluster for each pixel.
|Rp-Rc|+|Gp-Gc|+|Bp-Bc|

4.
Update cluster centers.

5.
Check convergence.

Exit Looping, if …
sum of the L1 distances between the new cluster centers and the previous cluster centers is less than epsilon*num_clusters.

K-means clustering



K-means clustering

1.
Make a copy of your original image.

2.
Initialize cluster centers.
double** centers

3.
Find closest cluster for each pixel.
|Rp-Rc|+|Gp-Gc|+|Bp-Bc|

Assign pixel color with the cluster center color it belongs to.

4.
Update cluster centers.

5.
Check convergence.

6.
Render result.

10/15 – 10/20

Task 1. With Random Seeds

2.
Initialize cluster centers.
double** centers

- void RandomSeedImage(double **image, int num_clusters)

- Choose a random number!
- Range of values: 0 ~ 255
- What you can do: rand() % 256
- Set epsilon = 30 (fine tune as necessary)

centers[i][j]:
,where i = 1 … number of clusters
j = 1 … 3 (R,G,B channels)

10/21 – 10/24

Task 2. With Pixel Seeds

2.
Initialize cluster centers.
double** centers

- void PixelSeedImage(double **image, int num_clusters)

- Choose a random pixel from image!
- Range of values:
  column = 0 ~ (imageWidth – 1)
  row = 0 ~ (imageHeight – 1)
- What you can use: rand() % something…
- Set epsilon = 30.

centers[i][j]:
,where i = 1 … number of clusters
j = 1 … 3 (R,G,B channels)

10/21 – 10/24

Task 2. With Pixel Seeds

2.
Initialize cluster centers.
double** centers

Choose a pixel and make its RGB values a seed if it is sufficiently different (dist(L1) >= 100) from already-selected seeds!

- void PixelSeedImage(double **image, int num_clusters)

- Choose a random pixel from image!
- Range of values:
  column = 0 ~ (imageWidth – 1)
  row = 0 ~ (imageHeight – 1)
- What you can use: rand() % something…
- Set epsilon = 30.

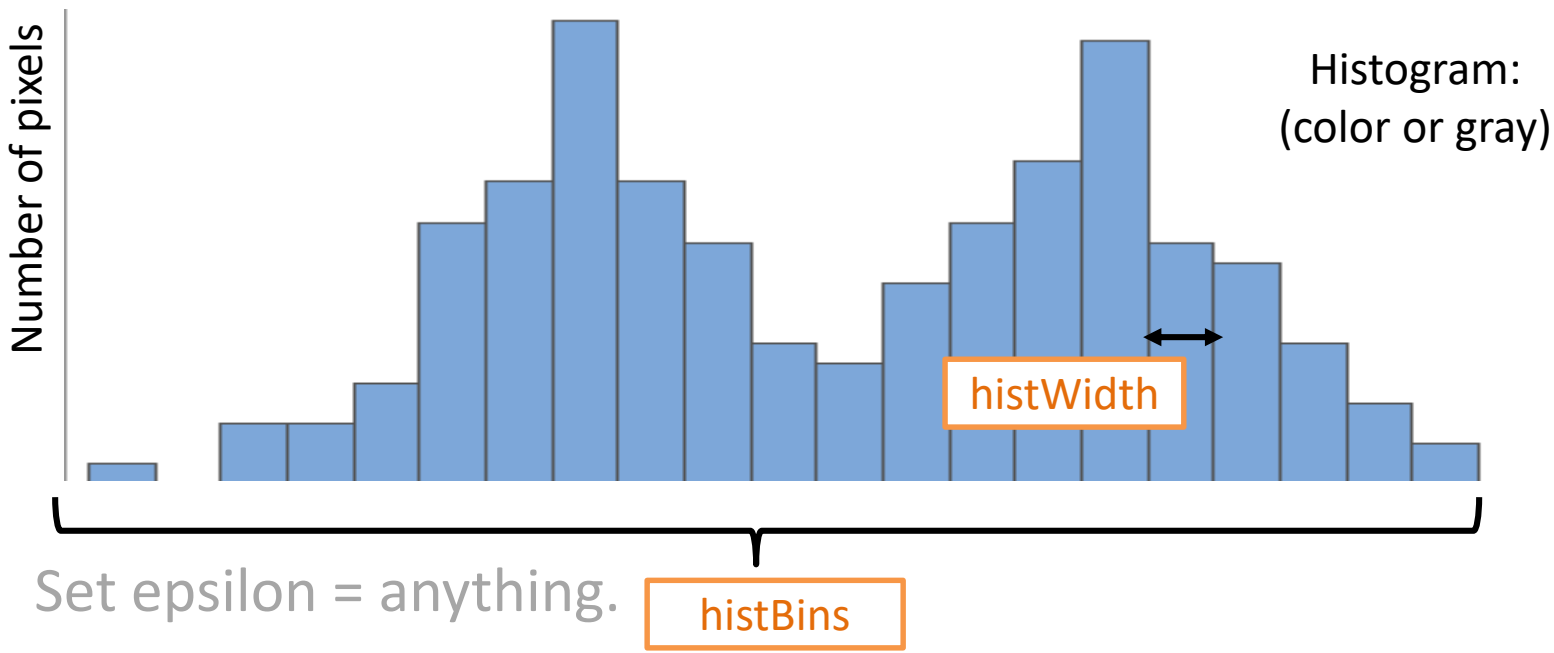
centers[i][j]:
,where i = 1 … number of clusters
j = 1 … 3 (R,G,B channels)

Extra Credit

Task 3. With Histogram Seeds

2.
Initialize cluster centers.
double** centers

- void HistogramSeedImage(double** image, int num_clusters)



- Set epsilon = anything.