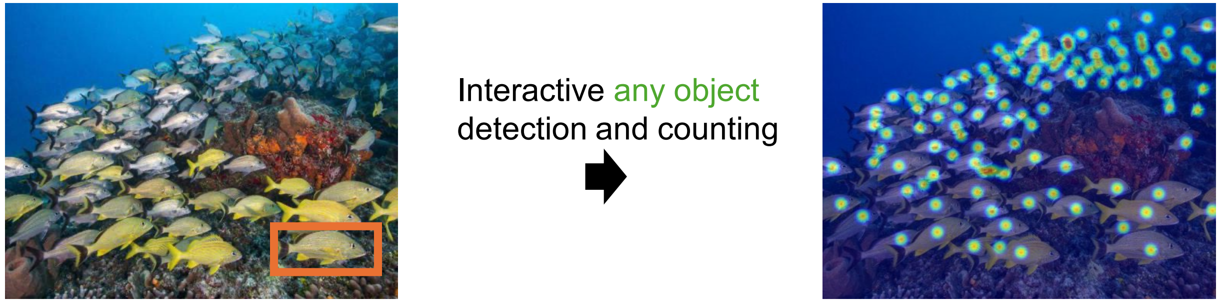
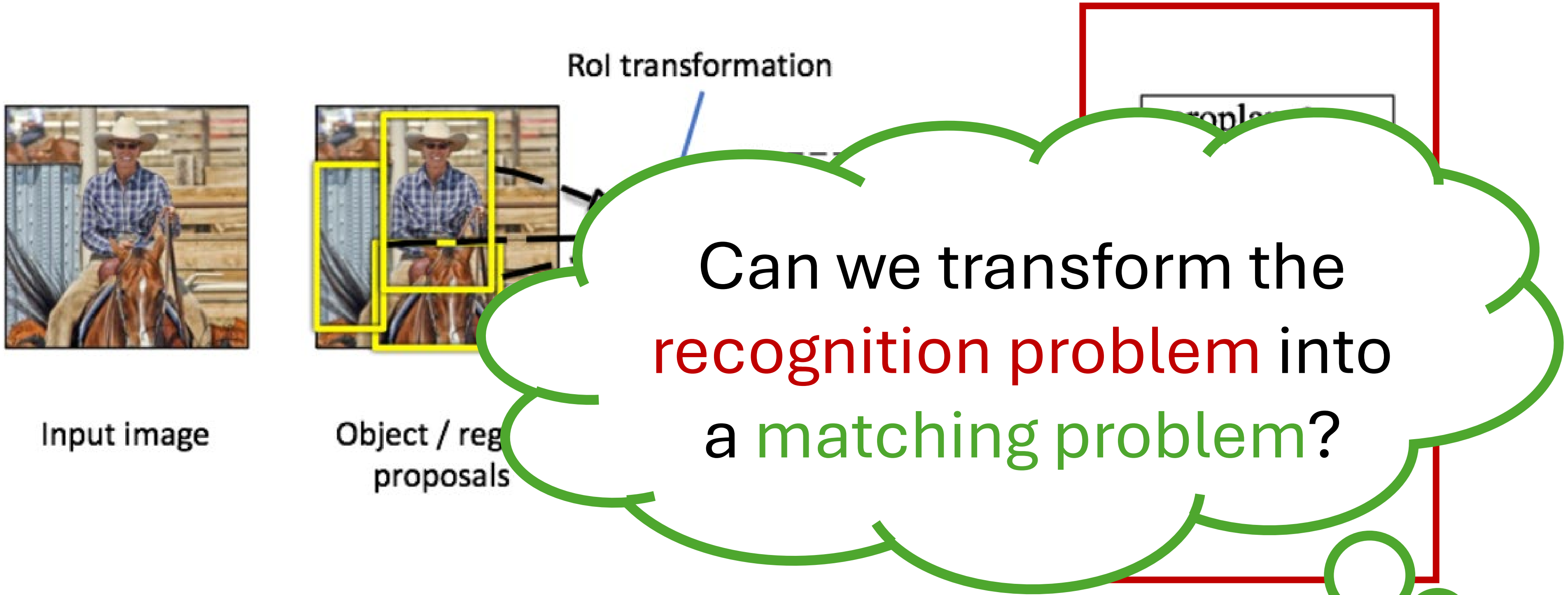


ConCoNet: Class-Agnostic Counting with Positive and Negative Exemplars

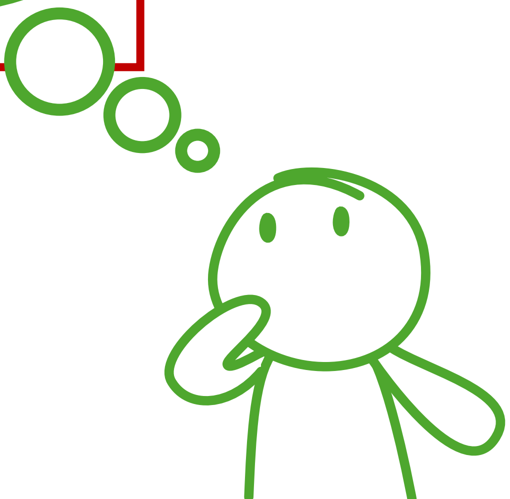
Adrienne Francesca O. Soliven, Daniel Stanley Tan, et al..

Standard detectors

Limited to a fixed set of object classes



Can we transform the recognition problem into a matching problem?

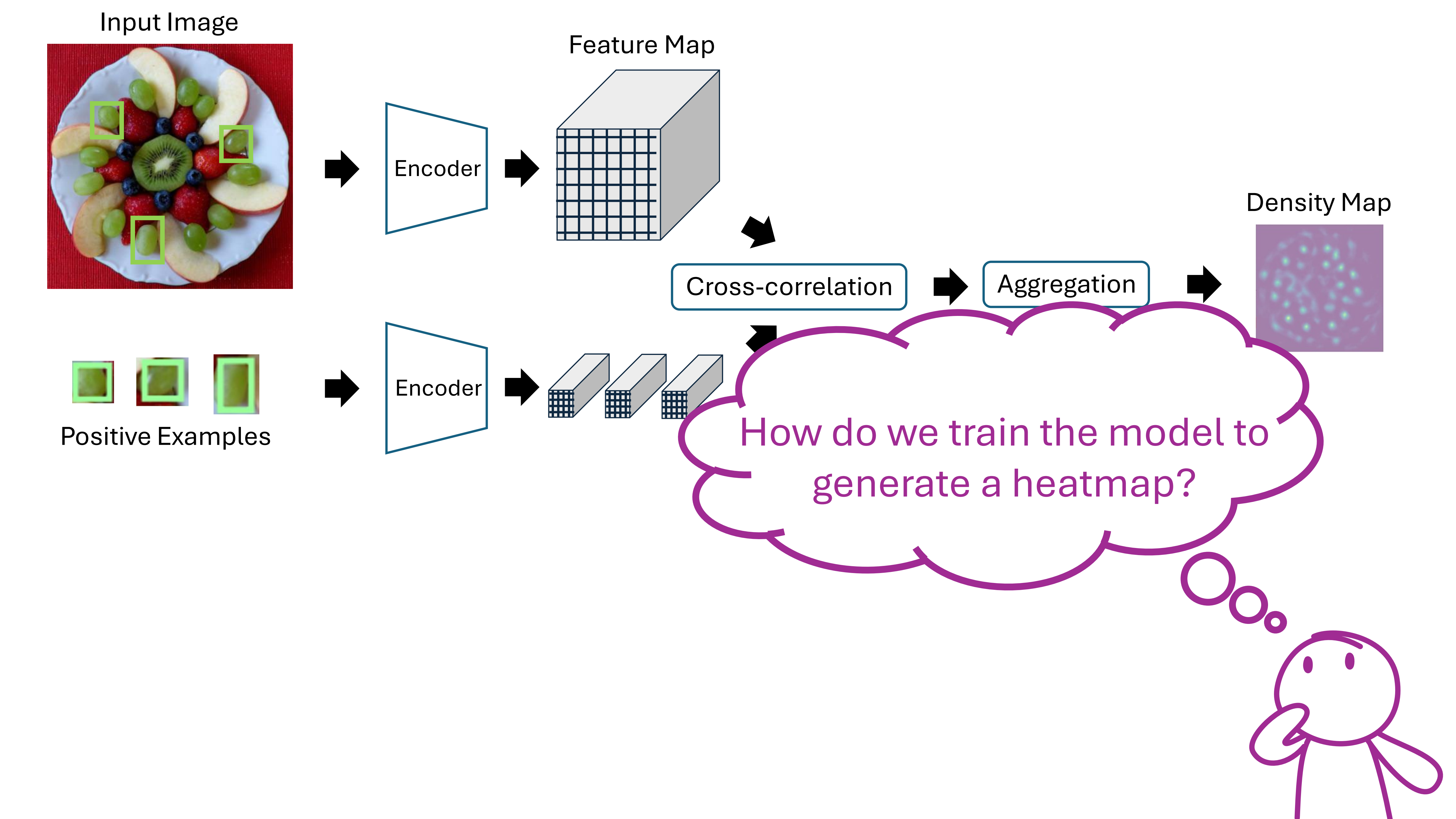


Recognition

What is this object?

Matching

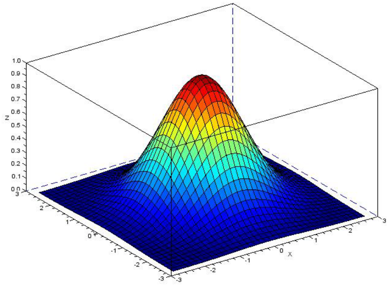
Are these two the same objects?

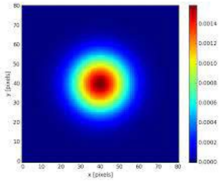

Input Image
Feature Map
Encoder
Encoder
Density Map
Cross-correlation
Aggregation
Positive Examples
How do we train the model to generate a heatmap?

Generating ground truth density maps

Input Image

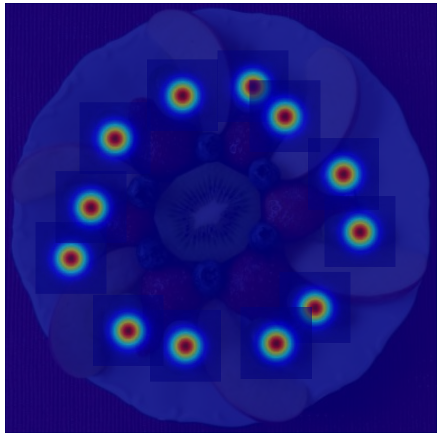
Add a Gaussian on every matching object





Sum = 1

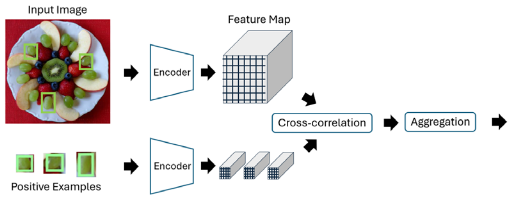
Density Map



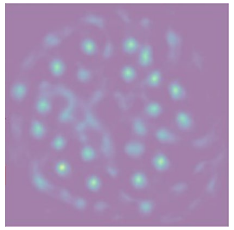
Sum = count

Mean Squared Error

Training the model



Heatmap



Handling overlapping gaussian points

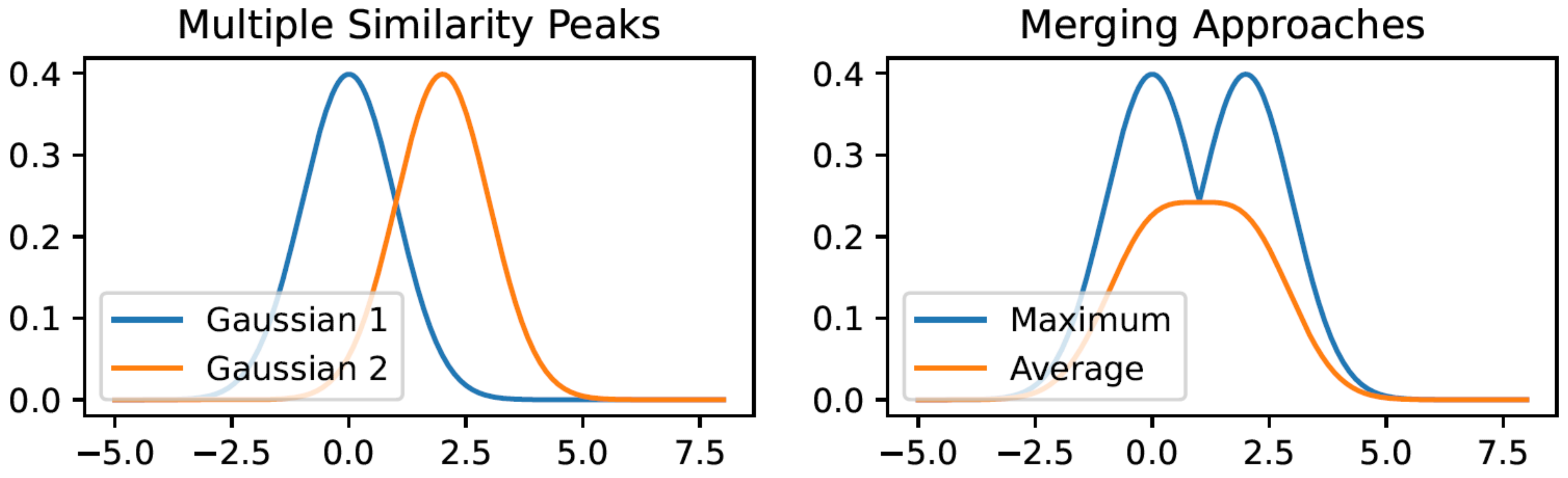
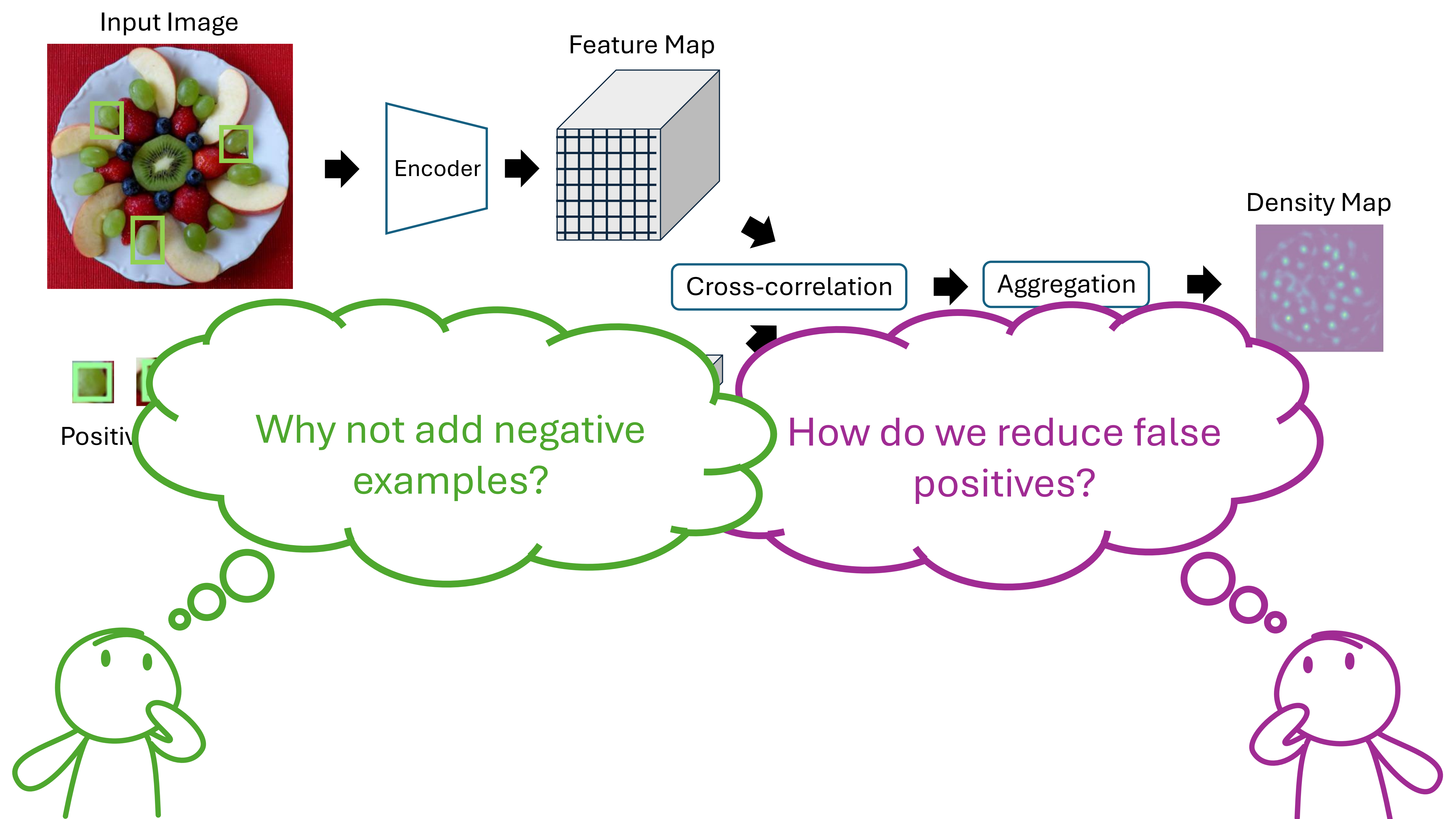
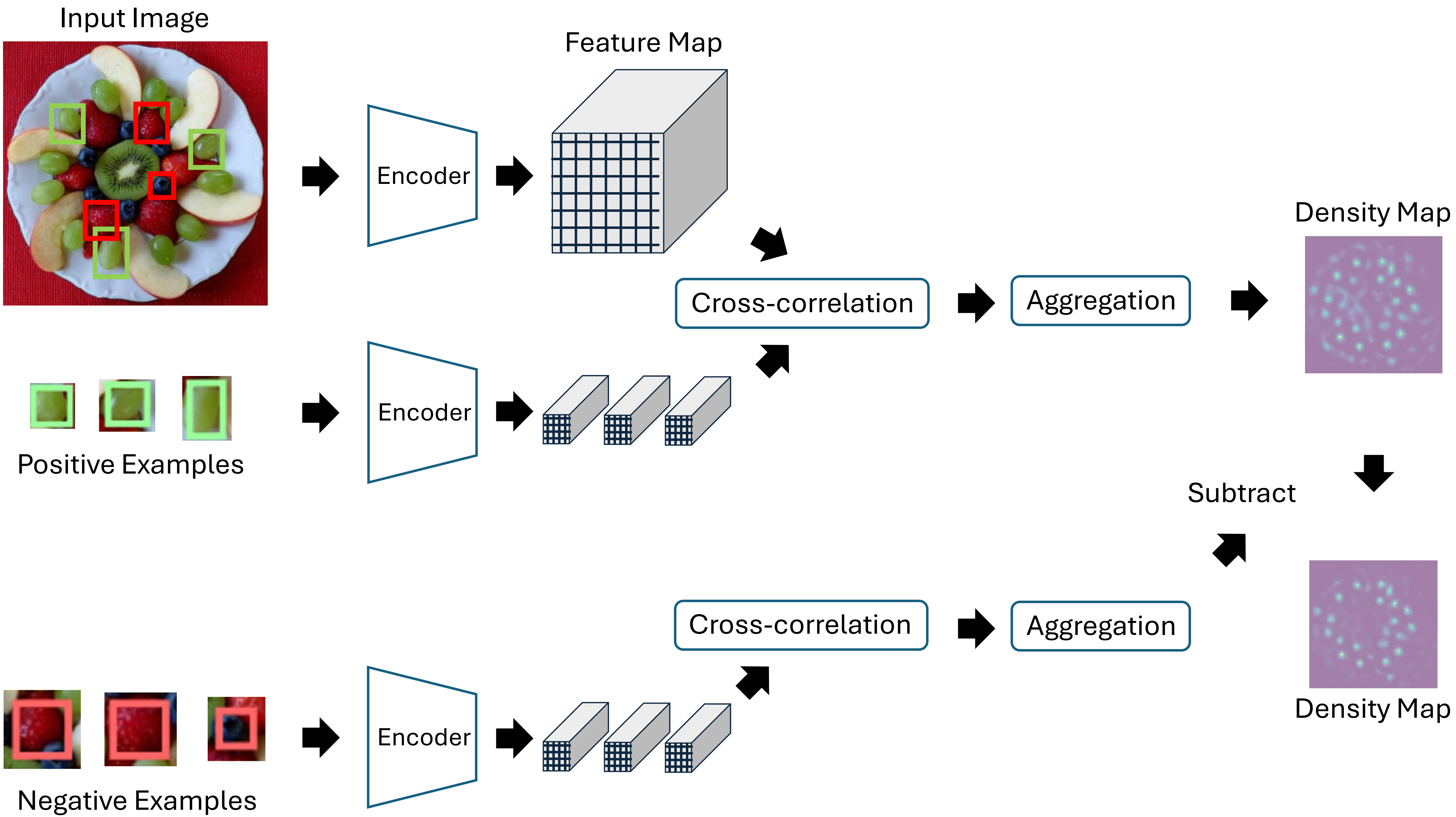
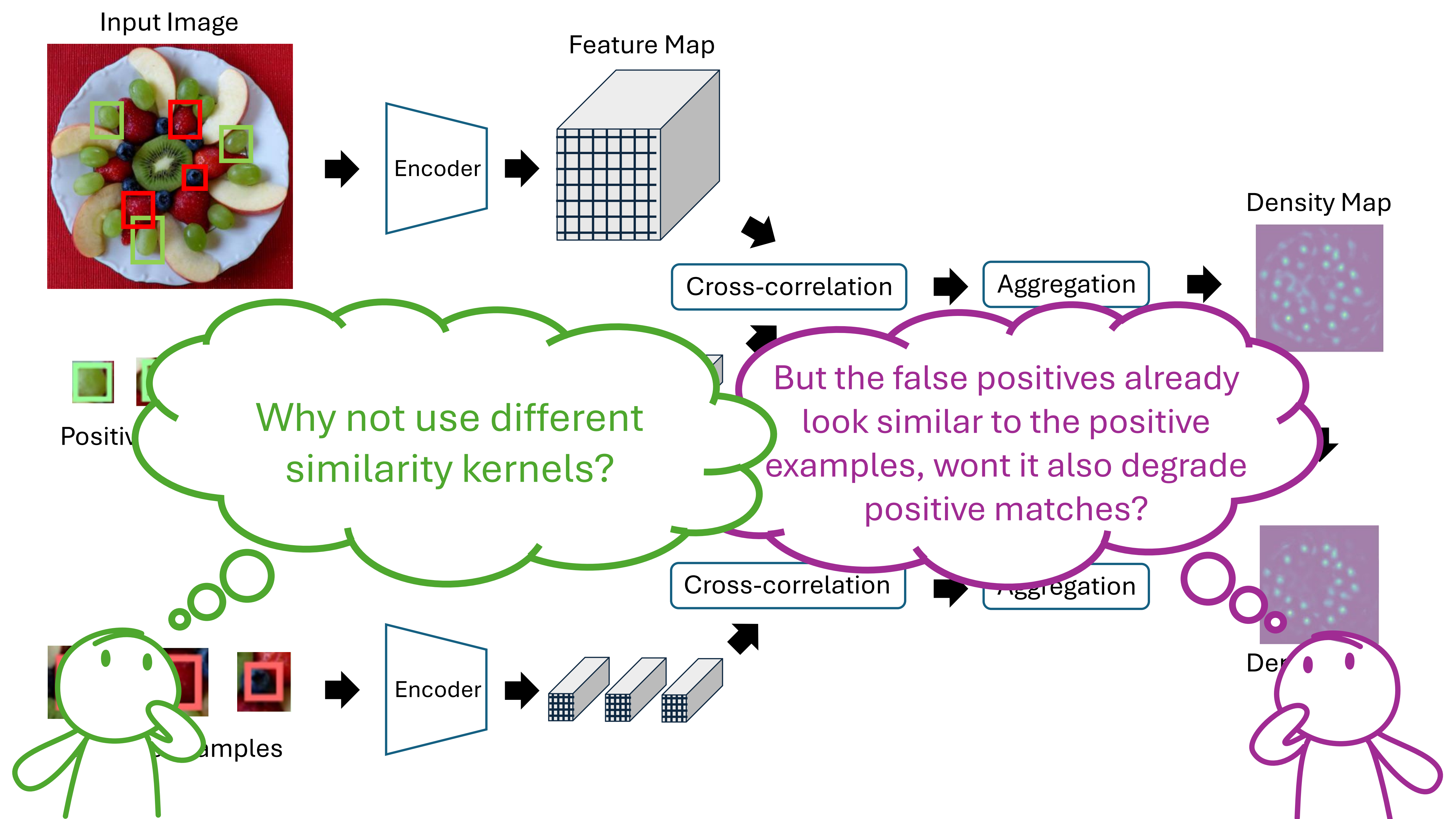


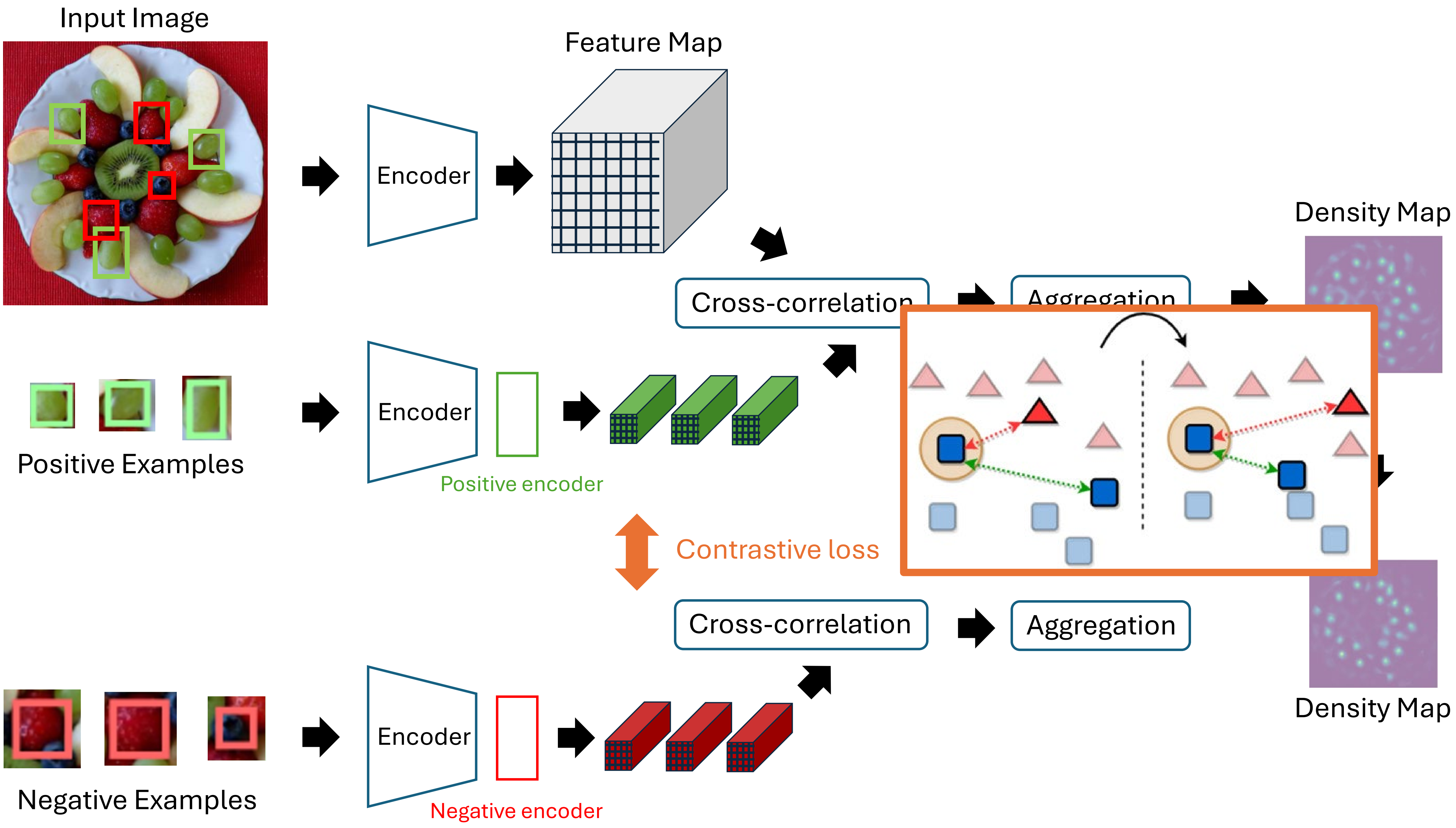
Fig. 3: Consider two different Gaussians (left) as a simplified similarity map. The visualization of merging the two (right) shows that the peaks are preserved using the max but not in the averaging.


Input Image
Feature Map
Encoder
Density Map
Cross-correlation
Aggregation
Positiv
Why not add negative examples?
How do we reduce false positives?


Input Image
Encoder
Feature Map
Density Map
Cross-correlation
Aggregation
Encoder
Positive Examples
Subtract
Cross-correlation
Aggregation
Density Map
Encoder
Negative Examples



Input Image
Feature Map
Encoder
Encoder
Density Map
Cross-correlation
Aggregation
Cross-correlation
Aggregation
Why not use different similarity kernels?
But the false positives already look similar to the positive examples, wont it also degrade positive matches?



Input Image
Feature Map
Encoder
Encoder
Encoder
Positive Examples
Positive encoder
Negative Examples
Negative encoder
Cross-correlation
Aggregation
Density Map
Contrastive loss
Cross-correlation
Aggregation
Density Map

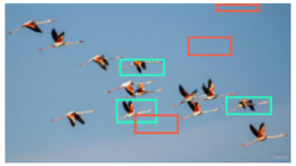
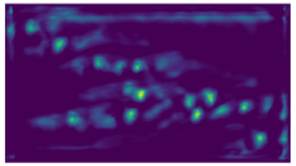
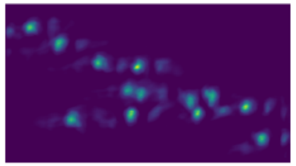
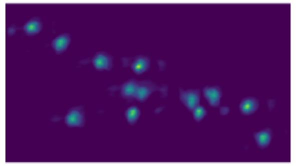
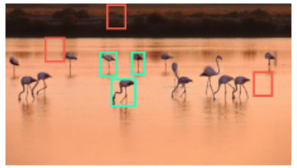
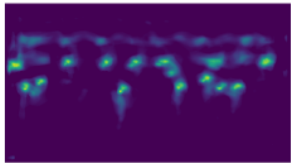
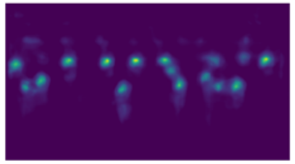
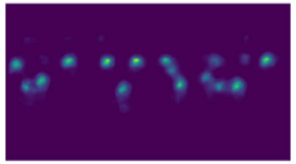
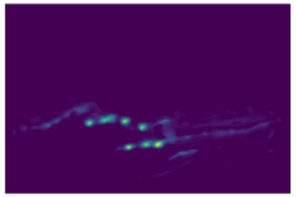
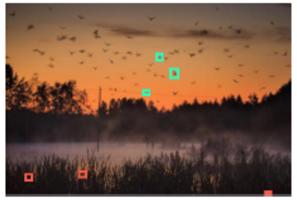
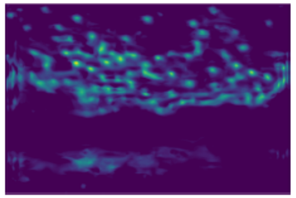
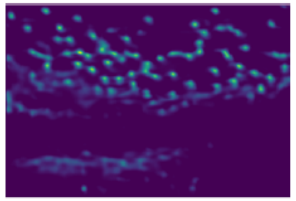
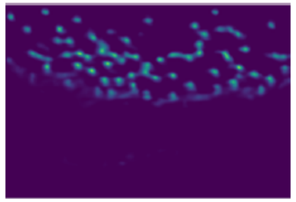
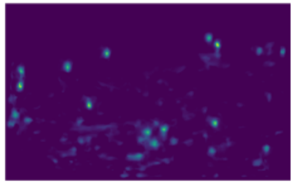
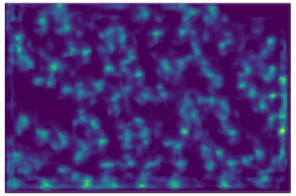
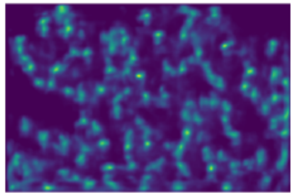
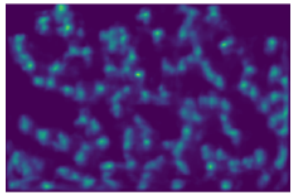
Input Image	Positives Only		Positives and Negatives
	FamNet [16]	ConCoNet (Ours)	ConCoNet (Ours)
GT Count: 13	Pred Count: 41.96	Pred Count: 18.28	Pred Count: 12.95
			
GT Count: 15	Pred Count: 31.68	Pred Count: 19.31	Pred Count: 14.62
			
GT Count: 9	Pred Count: 18.56	Pred Count: 13.22	Pred Count: 9.05
			
GT Count: 77	Pred Count: 170.16	Pred Count: 117.47	Pred Count: 71.49
			
GT Count: 11	Pred Count: 32.58	Pred Count: 19.34	Pred Count: 10.24
			
GT Count: 133	Pred Count: 166.06	Pred Count: 183.87	Pred Count: 134.66
			

Table 1: **Ablation study on components.** An analysis on the components of *ConCoNet* to evaluate the added performance of each combinations.

Components	Combinations				
Positive Kernel	✔	✗	✔	✔	✔
Negative Kernel	✗	✗	✗	✔	✔
Sim Fusion	✗	✔	✗	✔	✔
$\mathcal{L}_{posneg}$	✔	✗	✗	✔	✗
$\mathcal{L}_{Total}$	✗	✔	✔	✗	✔
MAE	102.38	29.65	21.23	20.89	18.77
RMSE	143.32	86.00	59.18	56.57	59.46

Table 2: **Few-shot methods comparison.** A quantitative comparison was performed between two simple baselines(Mean, Median), five main baselines (feature Reweighting FR few-shot detector, FSOD few-shot detector, GMN, MAML, FamNet), some are class-agnostic models and others are few-shot methods that are adapted and trained for counting. Results show that *ConCoNet* performed best in both the val set and test set of FSC-147.

	Val Set		Test Set	
Methods	MAE	RMSE	MAE	RMSE
Mean	53.38	124.53	47.55	147.67
Median	48.68	129.70	47.73	152.46
FR detector [23]	45.45	112.53	41.64	141.04
FSOD detector [24]	36.36	115.00	32.53	140.65
Pretrained GMN [17]	60.56	137.78	62.69	159.67
GMN [17]	29.66	89.81	26.52	124.57
MAML [25]	25.54	79.44	24.90	112.68
FamNet [16]	23.75	69.07	22.08	99.54
Ours (Positives Only)	21.23	**59.18**	22.34	**91.75**
Ours	**18.77**	59.46	**18.02**	93.63

Table 3: MAE performance of ConCoNet on the validation set as the number of positive and negative exemplars increases and decreases.

		# of Positives		
		1	2	3
# of Negatives	0	24.15	23.02	21.23
	1	20.27 ± 0.11	19.74 ± 0.32	19.13 ± 0.03
	2	19.55 ± 0.25	19.28 ± 0.20	18.98 ± 0.10
	3	19.21 ± 0.23	19.11 ± 0.56	**18.77** ± 0.20

Annotation tool

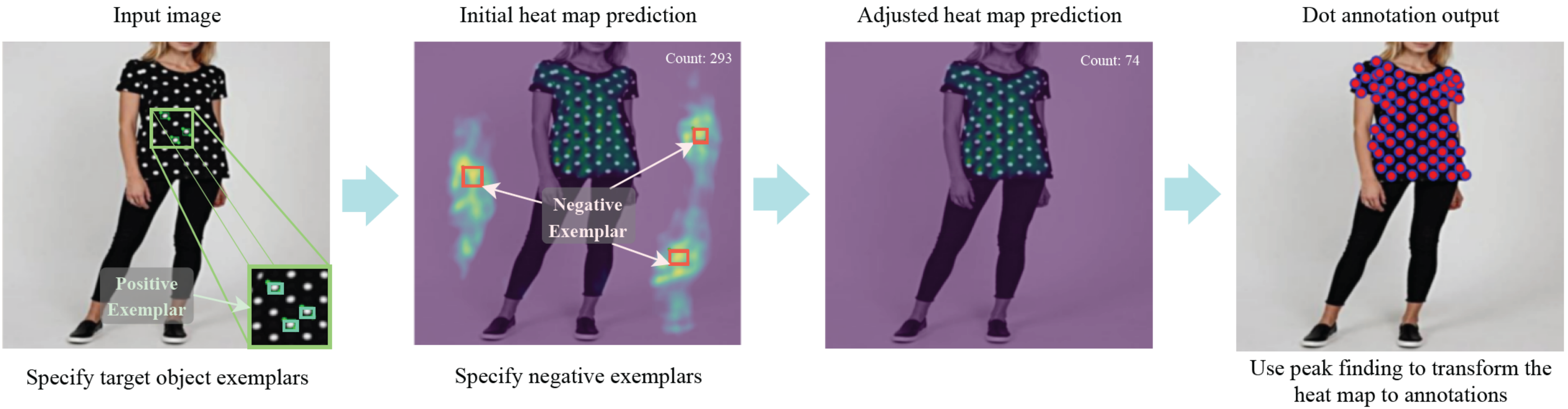
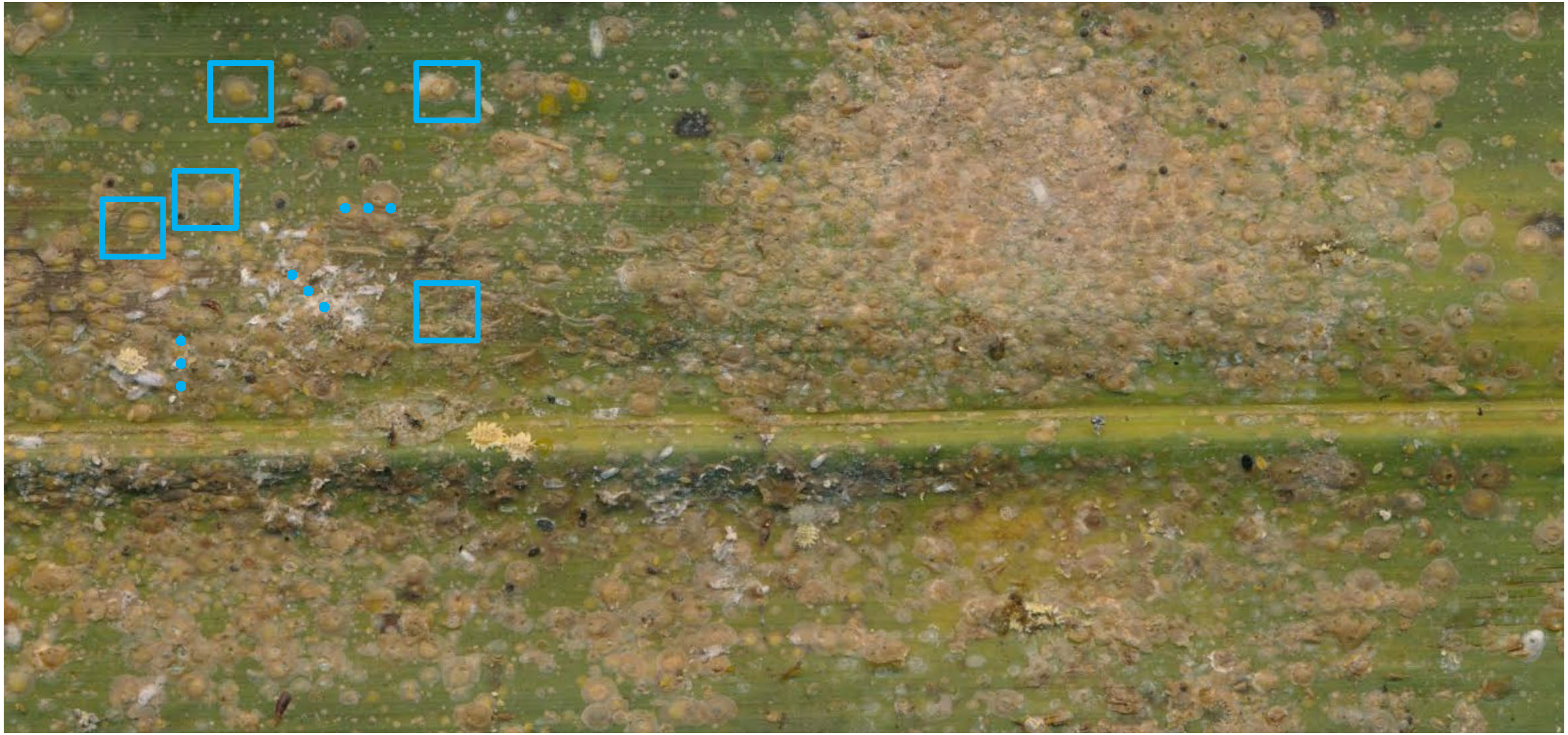


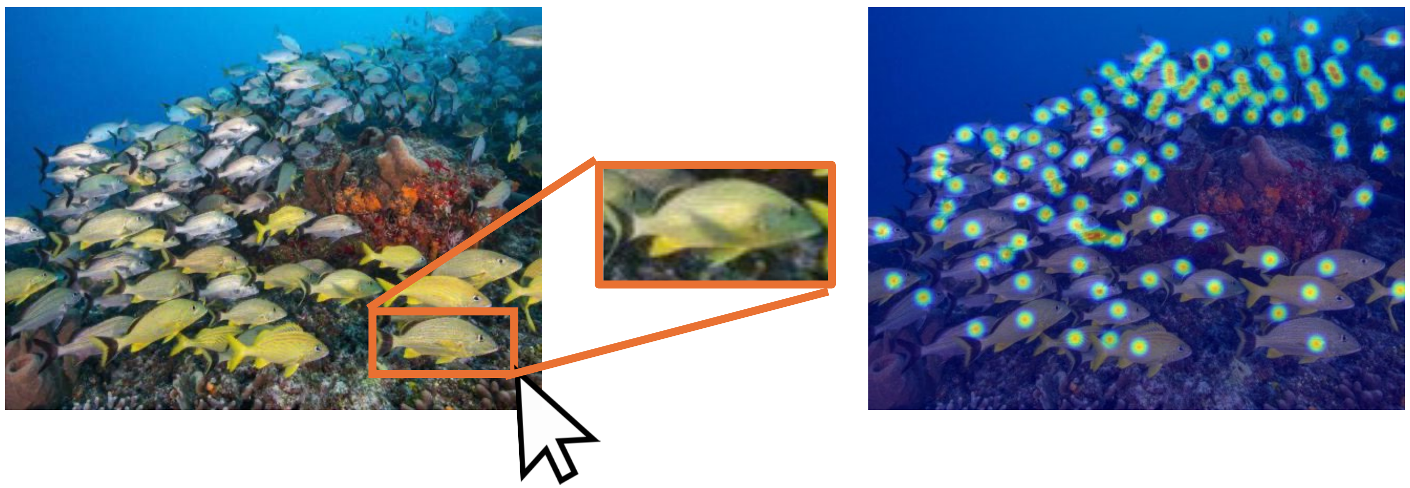
Fig. 4: **Semi-Automated Annotation Tool Process.** The first image is the image input and green boxes are the positive exemplars bounding boxes that specify examples of the target object. The second image is the initial heat map with the count prediction. The model experienced some confusion due to the visual similarity between the target object and background. The user inputs the negative exemplars (denoted by the red bounding boxes) to specify what should be ignored. The third image is an improved heat map based on the previous user input. The last image shows that the latest heat map is converted into a dot annotation.

Less lag time

- Can immediately start on a new domain
- Can reduce labeling efforts by propagating labels to similar objects



Interactive any object detection and counting



User-specified
target object

Thank you! Any Questions?