Localization in the Crowd with Topological Constraints

Shahira Abousamra[1], **Minh Hoai Nguyen**[1], **Dimitris Samaras**[1], **Chao Chen**[2]
[1]Stony Brook University, Department of Computer Science, USA
[2]Stony Brook University, Department of Biomedical Informatics, USA

Abstract

We address the problem of crowd localization, i.e., predicting dots corresponding to people in a crowded scene. Due to various challenges, a localization method is prone to spatial semantic errors, i.e., predicting multiple dots within a same person or collapsing multiple dots in a cluttered region. We propose a topological approach targeting these semantic errors. We introduce a topological constraint that teaches the model to reason about the spatial arrangement of dots. To enforce this constraint, we define a persistence loss based on the theory of persistent homology. The loss compares the topographic landscape of the likelihood map and the topology of the ground truth. Topological reasoning improves the quality of the localization algorithm especially near cluttered regions. On multiple public benchmarks, our method outperforms previous localization methods by a large margin. Additionally, we demonstrate the potential of our method in improving the performance in the crowd counting task.

Localization of people or objects, i.e., identifying the exact location of each instance, in a crowded scene is an important problem for many fields. Localization of people, animals, or biological cells provides detailed spatial information that can be crucial in journalism (Mcphail and McCarthy 2004), ecology (Elphick 2008) or cancer research (Barua et al. 2018). A high quality localization algorithm naturally solves the popular crowd counting problem, i.e., counting the number of people in a crowded scene (Idrees et al. 2018). Furthermore, the rich spatial information can be used in many other tasks, e.g., initialization of tracking algorithms (Ren et al. 2018), animal population study (Elphick 2008), tumor microenvironment analysis (Aukerman et al. 2020), and most recently, monitoring of social distancing (Yang et al. 2020).

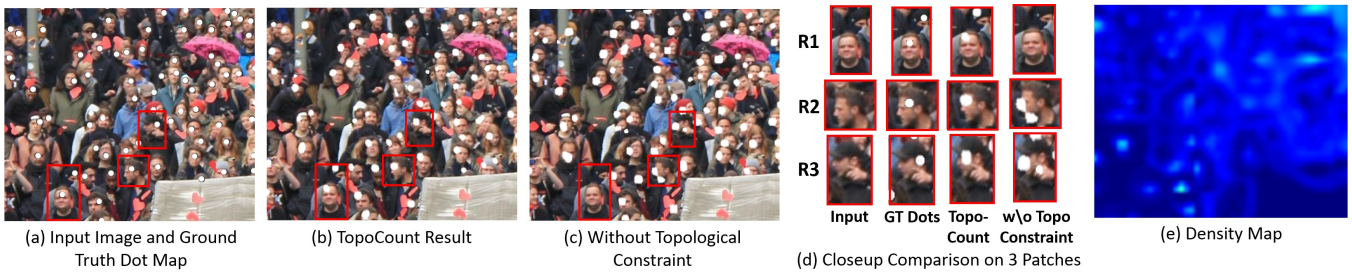
Despite many proposed methods (Zhao, Nevatia, and Wu 2008; Ge and Collins 2009; Liu, Weng, and Mu 2019; Babu Sam et al. 2020), localization remains a challenging task. Aside from fundamental challenges of a crowded scene such as perspective, occlusion, and cluttering, one key issue is the limitation of annotation. Due to the large number of target instances, the ground truth annotation is usually provided in the form of dots located inside the instances (Fig. 1(a)). These dots only provide limited information. A dot can be arbitrarily perturbed as long as it is within the target instance, which can be of very different scales. As a consequence, the features of dots are not specific. Without sufficient supervision, we cannot decide the boundary between instances. Thus, it is very hard to prevent spatial semantic errors, i.e., predicting multiple dots within a same person (false positives) or collapsing the dots of multiple persons in a cluttered area (false negatives).

In this paper, we propose a novel topological approach for the localization problem. We treat the problem as predicting a binary mask, called the Topological Map (Fig. 1(b)), whose connected components one-to-one correspond to the target dots. The number of components in the predicted mask should be the same as the number of ground truth dots. This spatial semantic constraint is indeed *topological*. During training we enforce such "topological constraint" locally, i.e., the topology should be correct within each randomly sampled patch. The topological constraint teaches the model to reason about spatial arrangement of dots and avoids incorrect phantom dots and dots collapsing. This significantly improves the quality of a localization method, especially near dense regions. See Fig. 1(b), (c) and (d).

To enforce the topological constraint, we introduce a novel loss, called *persistence loss*, based on the theory of persistent homology (Edelsbrunner and Harer 2010). Instead of directly computing the topology of the predicted binary mask, we inspect the underlying likelihood map, i.e., the sigmoid layer output of the neural network. The algorithm of persistent homology captures the topographic landscape features of the likelihood map, namely, modes and their saliency. Our persistence loss compares these modes and the true topology. Within any sample patch, if there are k true dots, the persistence loss promotes the saliency of the top k modes and penalizes the saliency of the remaining modes. This way it ensures that there are exactly k modes in the likelihood landscape, all of which are salient. A 0.5-thresholding of such a topologically correct likelihood map results in a binary mask with exactly k components, as desired.

We evaluate our method on various benchmarks and show that our proposed method, TopoCount, outperforms previous localization methods in various localization metrics.

Application to crowd counting. We further demonstrate the power of our localization method by applying it to a closely related problem, *crowd counting*. For the counting problem,



(a) Input Image and Ground Truth Dot Map (b) TopoCount Result (c) Without Topological Constraint (d) Closeup Comparison on 3 Patches (e) Density Map

Figure 1: (a) a sample image and the ground truth (GT) dots; (b) the localization result of our method (TopoCount); (c) without the topological constraints, topological errors may happen and the localization quality is impacted; (d) closeup view of some specific patches. Without topological constraint, the prediction often misses dots or collapses nearby dots (R1). It can also create phantom dots (R2 and R3). TopoCount successfully avoids such errors; (e) a density map from a SOTA counting method (Ma et al. 2019). The density map loses important topological information and cannot recover spatial arrangement of dots.

training images are also annotated with dots, but the task is simpler; one only needs to predict the total number of instances in each image. State-of-the-art (SOTA) counting algorithms, such as (Liu, Salzmann, and Fua 2019a; Ma et al. 2019; Jiang et al. 2020), learn to approximate a density map of the crowd whose integral gives the total count in the image. The learnt density function, even with accurate counting number, can significantly lose the topological characterization of the target density, especially near the dense population region. See Fig. 1(e) for an example.

We incorporate our localization result as complimentary information for density-based counting algorithms. By introducing our TopoCount results as additional input to SOTA counting algorithms (Ma et al. 2019; Liu, Salzmann, and Fua 2019a), we improve their counting performance by 7 to 28 % over different public benchmarks. This further demonstrates the power of the spatial configuration information that we obtain through the topological reasoning.

In summary, our technical contribution is three-fold.

- We propose a topological constraint to address the topological errors in crowd localization.
- To enforce the constraint, we propose a novel persistence loss based on the theory of persistent homology. Our method achieves SOTA localization performance.
- We further integrate the topology-constrained localization algorithm into density-based counting algorithms, to achieve a SOTA counting method.

1 Related Work

We discuss various localization approaches; some of which learn the localization algorithm jointly with the counting model. Babu Sam et al. (2020) learn to predict bounding boxes of human heads by fusing multi-scale features. The model is trained with cross entropy loss over the whole image and special focus on selected high error regions. Liu et al. (2018) performs detection using Faster RCNN (Ren et al. 2015). Faster RCNN has been shown to not scale well with the increasing occlusion and clutter in crowd counting benchmarks (Wang et al. 2020). Liu, Weng, and Mu (2019) also learn to predict localization map as a binary mask. They use a weighted cross-entropy loss to compensate for the unbalanced foreground/background pixel populations. In dense regions, the localization is further improved by recurrent zooming. However, all these methods are not explicitly modeling the topology of dots and thus cannot avoid topological errors (phantom dots and dots collapsing) as our method does.

A related method is (Laradji et al. 2018). It formulates the problem as a semantic segmentation problem. Blobs of the segmentation mask correspond to the target object instances. A blob is split if it contains multiple true dots and is suppressed if it does not contain any true dot. This method is not robust to the perturbation of dot locations; a blob that barely misses its corresponding true dot will be completely suppressed. On the contrary, our method leverages the deformation-invariance of topological structures arising from dots, and thus can handle the dot perturbation robustly.

SOTA counting methods are based on density function estimation (Ranjan, Le, and Hoai 2018; Cao et al. 2018; Li, Zhang, and Chen 2018; Liu, Salzmann, and Fua 2019b; Ma et al. 2019; Jiang et al. 2020). These methods train a neural network to generate a density function, the integral of which represents the estimated object count (Lempitsky and Zisserman 2010). The ground truth density functions are generated by Gaussian kernels centered at the dot locations. While these methods excel at counting, the smoothed density functions lose the detailed topological information of the original dots, especially in dense areas (see Fig. 1(e)). As a consequence, localization maps derived from the estimated density maps, e.g., via integer programming (Ma, Lei Yu, and Chan 2015) or via multi-scale representation of the density function (Idrees et al. 2018), are also of limited quality.

Topological information has been used in various learning and vision tasks. Examples include but are not limited to shape analysis (Reininghaus et al. 2015; Carriere, Cuturi, and Oudot 2017), graph classification (Hofer et al. 2017; Zhao and Wang 2019), clustering (Ni et al. 2017; Chazal et al. 2013) and image segmentation (Mosinska et al. 2018; Chan et al. 2017; Waggoner et al. 2015). Persistent-homology-based objective functions have been used for image segmentation (Hu et al. 2019; Clough et al. 2019), graphics (Poulenard, Skraba, and Ovsjanikov 2018) and machine learning model regularization (Hofer et al. 2019; Chen

et al. 2019). To the best of our knowledge, our method is the first to exploit topological information in a crowd localization and counting task, and to use a topology-informed loss to solve the corresponding topological constraint problem.

2 Method: TopoCount

We formulate the localization problem as a structured prediction problem. Given training images labeled with *dot annotations*, i.e., sets of dots representing persons (Fig. 1(a)), we train our model to predict a binary mask. Each connected component in the mask represents one person. We take the centers of the connected components as the predicted dots. For training, we expand the dot annotations of training images into dot masks by a slight dilation of each dot, but with the condition that the expanded dots do not overlap. We call this "dot mask" the *ground truth dot map*.

To train a model to predict this binary ground truth dot map, we use a U-Net type architecture with a per-pixel loss. The output of the model after the Sigmoid activation is called the *topological likelihood map*. During inference, a final thresholding step is applied to the likelihood to generate the binary mask. We call the mask the *topological dot map* as it is required to have the same topology as the ground truth dot map. Fig. 5 shows our overall architecture. The rest of this section is organized as follows. We first introduce the topological constraint for the topological dot map. Next, we formalize the persistence loss that is used to enforce the topological constraint. Afterwards, we provide details of the architecture and training. Finally, we discuss how to incorporate our method into SOTA counting algorithms to improve their counting performance significantly.

2.1 Topological Constraint for Crowd Localization

For the localization problem, a major challenge is the perturbation of dot annotation. In the training dot annotation a dot can be at an arbitrary location of a person and can correspond to different parts of a human body. Therefore it is hard to control the spatial arrangement of the predicted dots. As illustrated in Fig. 1(d), a model without special design can easily predict multiple "phantom dots" at different body parts of the same person. At cluttered regions, the model can exhibit "dot collapsing". To address these semantic errors, we must teach the model to learn the spatial contextual information and to reason about the interactions between dots. A model needs to know that nearby dots are mutually exclusive if there are no clear boundary between them. It should also encourage more dots at cluttered regions. To teach the model this spatial reasoning of dots, we define a *topological constraint* for the predicted topological dot map, y:

Topological constraint for localization. *Within any local patch of size $h \times w$, the Betti number of dimension zero, i.e. the number of connected components, of y equals to the number of ground truth dots.*

This constraint allows us to encode the spatial arrangement of dots effectively without being too specific about their locations. This way the model can avoid the topological errors such as phantom dots and dots collapsing, while being robust to perturbation of dot annotation. Next, we introduce a novel training loss to enforce this topological constraint.

2.2 Persistence Loss

Directly enforcing the topological constraint in training is challenging. The number of connected components and the number of dots within each patch are discrete values and their difference is non-differentiable. We introduce a novel differentiable loss called *persistence loss*, based on the persistence homology theory (Edelsbrunner, Letscher, and Zomorodian 2000; Edelsbrunner and Harer 2010). The key idea is that instead of inspecting the topology of the binary topological dot map, we use the continuous-valued likelihood map of the network, f. We consider f as a terrain function and consider the landscape features of the terrain. These features provide important structural information. In particular, we focus on modes, i.e., local maxima, of f. As illustrated in Fig. 2(b)(c), a salient mode of f, after thresholding, will become a connected component in the predicted topological dot map. A weak mode will miss the cutoff value and disappear in the dot map.

The persistence loss captures the saliency of modes and decides to enhance/suppress these modes depending on the ground truth topology. Given a patch with c many ground truth dots, our persistence loss enforces the likelihood f to only have c many salient modes, and thus c connected components in y. It reinforces the total saliency of the top c modes of f, and suppresses the saliency of the rest. The saliency of each mode, m, is measured by its *persistence*, $\mathrm{Pers}(m)$, which will be defined shortly. As an example, in Fig. 2(c), f has 5 salient modes. if $c = 4$, the persistence loss will suppress the mode with the least persistence -in this case $m5$- and will reinforce the other 4 modes. As a consequence, the mode $m2$ is enhanced and is separated from $m1$, avoiding a mode collapsing issue. Formally:

Definition 1 (Persistence Loss) *Given a patch, δ, with c ground truth dots, denote by $\mathcal{M}_c$ the top c salient modes, and $\overline{\mathcal{M}_c}$ the remaining modes of f. The persistence loss of f at the patch δ is*

$$L_{Pers}(f, \delta) = -\sum\nolimits_{m \in \mathcal{M}_c} \mathrm{Pers}(m) + \sum\nolimits_{m \in \overline{\mathcal{M}_c}} \mathrm{Pers}(m) \tag{1}$$

Minimizing this loss is equivalent to maximizing the saliency of the top c modes and minimizing the saliency of the rest. Consequently, the function will only have c salient modes, corresponding to c components in the predicted mask, Fig. 2(e)(f). Next we formalize the mode saliency, called *persistence*, and derive the gradient of the loss.

Saliency/persistence of a mode. For a mode, m (local maximum), its *basin of attraction* is the region of all points from which a gradient ascent will converge to m. Intuitively, the persistence of m, measuring its "relative height", is the difference between its height, $f(m)$, and the level, $f(s)$, at which its basin of attraction meets that of another higher mode. See Fig. 2(g) for an illustration.

In implementation, the saliency/persistence of each mode is computed by capturing its local maximum and corresponding saddle point. To find each mode m_i and its cor-

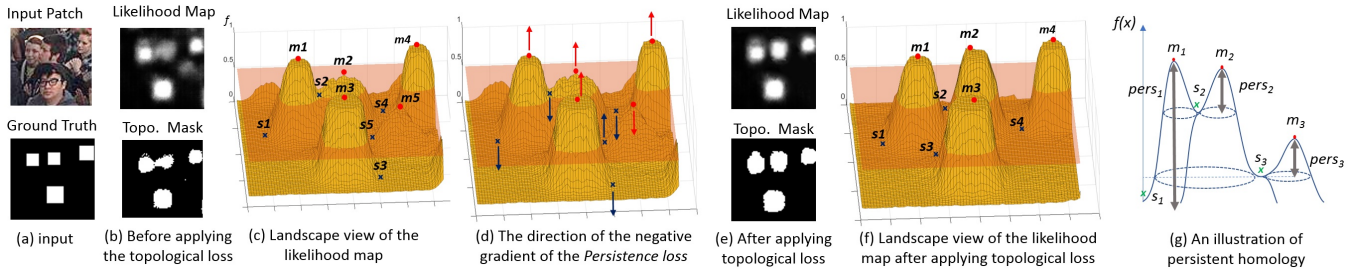


Figure 2: (a) An example image patch and its ground truth dot map (with 4 true dots). (b) The likelihood map and prediction mask without topological constraint. (c) A landscape view of the likelihood function. There are 5 modes (red dots) and their paired saddles(blue crosses). The top 4 salient ones ($m1$ to $m4$) are matched to the ground truth dots. The 5th, $m5$, is not matched. The thresholding excludes the weak mode ($m5$) in the predicted mask. But $m1$ and $m2$ are merged in the thresholded result because the saddle point between them ($s2$) is above the cutoff value. (d) Optimizing the Persistence loss will suppress $m5$ by reducing $f(m5)$. Meanwhile, it will enhance the saliency of $m2$ by increasing $f(m2)$ and decreasing $f(s2)$. When $f(s2)$ is below the threshold, $m1$ and $m2$ are separated into two components in the final prediction. (e) The likelihood and the prediction mask with topological constraint (persistence loss). The collapsing of $m1$ and $m2$ is avoided. (f) A landscape view of the likelihood function in (e). Only 4 modes remain. (g) An illustration of persistent homology. Three modes are paired with saddles at which their attractive basins merge with others. The differences $f(mi) - f(si)$, $i = 1, 2, 3$ are their persistence.

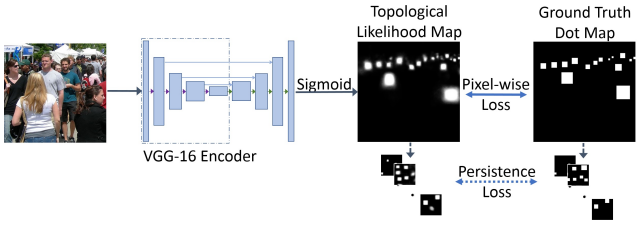


Figure 3: TopoCount has a U-Net style architecture with a VGG-16 encoder. During training a pixel-wise loss (DICE loss) is applied on the whole image and persistence loss is applied on sampled patches.

responding saddle point s_i where the component of m_i dies, we use a merging tree algorithm (Edelsbrunner and Harer 2010; Ni et al. 2017). A pseudo code of the algorithm is outlined in Appendix A.

This algorithm is almost linear. The complexity is $O(n \log n + n\alpha(n))$, where n is the patch size. The $O(n \log n)$ term is due to the sorting of all pixels. $O(n\alpha(n))$ is the complexity for the union-find algorithm for merging connected components. $\alpha(n)$ is the inverse Ackermann's function, which is almost constant in practice. The algorithm will detect all critical points, i.e. modes and saddle points, at different thresholds and pair them properly corresponding to all topological features of the function/landscape.

Having obtained the critical points of the likelihood function using the above algorithm, we apply the persistence loss as follows: For each component c_i, denote by m_i its birth maximum and by s_i its death saddle critical points. The persistence of c_i is $\text{Pers}(m_i) = f(m_i) - f(s_i)$. We sort all modes (or maximum-saddle pairs) according to their persistence. Denote by $\mathcal{M}_c$ the set of the top c salient modes, and by $\overline{\mathcal{M}_c}$ the remaining modes. The persistence loss in Equation (1) can be rewritten as

$$L_{Pers}(f, \delta) = -\sum_{m_i \in \mathcal{M}_c} (f(m_i) - f(s_i)) + \sum_{m_i \in \overline{\mathcal{M}_c}} (f(m_i) - f(s_i)) \quad (2)$$

When we take the negative gradient of the loss, for each of the top c modes, we will improve its saliency by increasing the function value at the maximum, $f(m_i)$, and decreasing the function value at its saddle $f(s_i)$. But for each other mode that we intend to suppress, the negative gradient will suppress the maximum's value and increase the saddle point's value. An important assumption in this setting is that the critical points, m_i and s_i, are constant when taking the gradient. This is true if we assume a discretized domain and a piecewise linear function f. For this discretized function, within a small neighborhood, the ordering of pixels in function value f remains constant. Therefore the algorithm output of the persistent computation will give the same set of mode-saddle pairs. This ensures that s_i and m_i's for all modes remain constant. The gradient of the loss w.r.t. the network weights, W, $\nabla_W L_{Pers}(f, \delta) = -\sum_{m_i \in \mathcal{M}_c} \left(\frac{\partial f(m_i)}{\partial W} - \frac{\partial f(s_i)}{\partial W} \right) + \sum_{m_i \in \overline{\mathcal{M}_c}} \left(\frac{\partial f(m_i)}{\partial W} - \frac{\partial f(s_i)}{\partial W} \right)$.

2.3 TopoCount: Model Architecture and Training

TopoCount computes the topological map that has the same topology as the dot annotation. To enable the model to learn to predict the dots quickly, we provide per-pixel supervision using DICE loss (Sudre et al. 2017). The DICE loss (L_{DICE}) given Ground truth (G) and Estimation (E) is: $L_{DICE}(G, E) = 1 - 2 \times \frac{(\sum G \circ E) + 1}{(\sum G^2 + \sum E^2) + 1}$, where $\circ$ is the Hadamard product.

Formally, the model is trained with the loss:

$$L = L_{DICE} + \lambda_{pers} L_{Pers} \quad (3)$$

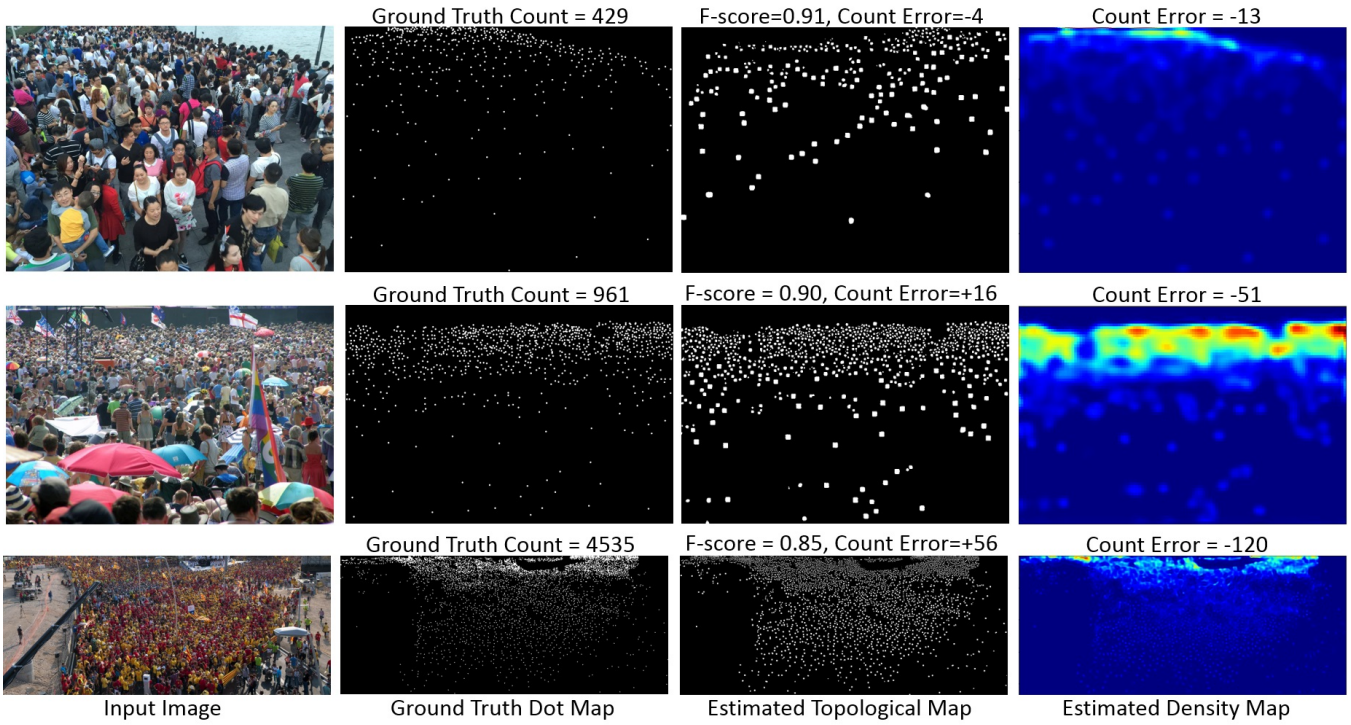


Figure 4: Sample results from different density crowd images. The columns represent the original image, ground truth, topological map by TopoCount, and the estimated density map by the integration of Bayesian + TopoCount.

in which λ_{pers} adjusts the weight of the persistence loss. An ablation study on the weight λ_{pers} is reported in the experiments. To provide more balanced samples for the per-pixel loss, we dilate the original dot annotation (treated as a supervision mask) slightly, but ensure that the dilation does not change its topology. The masks of two nearby dots stop dilating if they are about to overlap and impose false topological information. The size of the dilated dots is not related to the scale of the objects. These dilated dot masks from ground truth are used for training. Note that the persistence loss is applied to the likelihood map of the model, f.

Model Architecture Details. We use a UNet (Ronneberger, P.Fischer, and Brox 2015) style architecture with a VGG-16 encoder (Simonyan and Zisserman 2015). The VGG-16 backbone excludes the fully connected layers and so the backbone has around 15 million trainable parameters. There are skip connections between the corresponding encoding and decoding path blocks at all levels beyond the first. The skip connection between the first encoder block and last decoder block is pruned to avoid overfitting on low level features -such as simple repeated patterns that often occur in crowd areas. The final output is the raw topological map, a Sigmoid activation is applied to generate the likelihood map. More architecture details are in Appendix B. The model is shown in Fig 5.

2.4 Integration with Counting Methods

Density map based counting methods operate on a higher level of abstraction compared to crowd localization techniques. When there is a lot of ambiguity, such as in extreme dense far away populations, density map counting methods can provide better overall count estimate. However, we argue that high quality localization maps provide additional spatial configuration information that can improve counting algorithms. We propose a simple procedure that can integrate TopoCount with many counting methods. As a proof-of-concept, we integrate TopoCount with 2 popular density-based counting methods: Bayesian (Ma et al. 2019) and CAN (Liu, Salzmann, and Fua 2019a). We will show that the integration of the localization result learnt with topological constraint significantly boosts the performance of SOTA counting algorithms (7 to 28%, see Section 3, Experiments). This further demonstrates the power of the spatial configuration information we obtain through the topological reasoning.

For the integration, we use a pre-trained TopoCount model. The raw output dot map and topological likelihood map are concatenated to the RGB image as the input of a density estimation model. The intuition is that the topological map will act as a prior and will guide the network through implicit attention. The model has to be adjusted to account for the change in the number of input channels. Both networks: Bayesian (Ma et al. 2019) and CAN (Liu, Salzmann, and Fua 2019a), are partially initialized with pre-trained VGG models (Simonyan and Zisserman 2015), except for the input layer. We modify the first layer's input layer and randomly initialize its weights. The network is trained end-to-end.

	ShanghaiTech A			ShanghaiTech B			UCF QNRF		
Model	G(1)	G(2)	G(3)	G(1)	G(2)	G(3)	G(1)	G(2)	G(3)
CSRNet (Li, Zhang, and Chen 2018)	76	113	149	13	21	29	157	187	219
Bayesian (Ma et al. 2019)	75	90	130	**10**	**14**	23	**100**	**117**	150
LSC-CNN (Babu Sam et al. 2020)	70	95	137	**10**	17	27	126	160	206
TopoCount (ours)	**69**	**81**	**104**	**10**	**14**	**20**	102	119	**148**

Table 1: Grid Average Mean absolute Errors (GAME)

Method	Prec.	Recall	F-score
MCNN	59.93%	63.50%	61.66%
CL-CNN D_∞	75.8%	59.75%	66.82%
LSC-CNN	74.62%	73.50%	74.06%
TopoCount	**81.77%**	**78.96%**	**80.34%**

Table 2: Localization accuracy on the UCF QNRF dataset, with metric in (Idrees et al. 2018)

Method	F1-m / Pre / Rec (%)
Faster RCNN (Ren et al. 2015)	σ_l : 6.7 / **95.8** / 3.5 σ_s : 6.3 / **89.4** / 3.3
TinyFaces (Hu and Ramanan 2017)	σ_l : 56.7 / 52.9 / 61.1 σ_s : 52.6 / 49.1 / 56.6
VGG+GPR (Gao et al. 2019)	σ_l : 52.5 / 55.8 / 49.6 σ_s : 42.6 / 45.3 / 40.2
RAZ_Loc (Liu, Weng, and Mu 2019)	σ_l : 59.8 / 66.6 / 54.3 σ_s : 51.7 / 57.6 / 47.0
TopoCount (ours)	σ_l : **69.1** / 69.5 / **68.7** σ_s : **60.1** / 60.5 / **59.8**

Table 3: NWPU-Crowd Localization Challenge

	BCE	DICE	$\lambda = 0.5$	$\lambda = 1.0$	$\lambda = 1.5$	$\lambda = 2.0$
G(3)	122	114	109	**104**	**104**	107

Table 4: Ablation study for the loss function. Compare the localization score G(3) on the ShanghaiTech A dataset of TopoCount trained with: (a) weighted BCE loss, (b) DICE loss ($\lambda = 0$), (c) DICE and Persistence loss ($\lambda \in [0.5, 2]$).

3 Experiments

We validate our method on popular and large scale crowd counting benchmarks including ShanghaiTech parts A and B (Zhang et al. 2016), UCF CC 50 (Idrees et al. 2013), UCF QNRF (Idrees et al. 2018), JHU++ (Sindagi, Yasarla, and Patel 2020), and NWPU Challenge (Wang et al. 2020).

For the localization task, our method is superior compared to other methods. Moreover, we show that the localization results of our method benefits the counting task.

Training Details. We train our *TopoCount* with the dilated ground truth dot mask. The dilation is by default up to 7 pixels. For JHU++ and NWPU, which are provided with head box annotation, we use a more accurate dilation guided by the box size, $\max$(7, box width/2, box height/2). In all cases the dilation is no more than half the distance to the nearest neighbor to avoid overlapping of nearby dots.

The window size of the patch for topological constraint controls the level of localization we would like to focus on. Since the scale of persons within an image is highly heterogeneous, varying the window size based on scale sounds intriguing. However the ground truth dot annotation generally do not carry scale information. As a result, we fix the patch size for each dataset. We use 50×50 pixels patches for ShanghaiTech and UCF CC 50, and 100×100 pixels patches for the larger scale datasets UCF QNRF, JHU++, and NWPU to account for the larger scale variation. An ablation study on the patch size selection is in Appendix C.3. The persistence loss is applied on grid tiles to enforce topological consistency between corresponding prediction and ground truth tiles/patches. Coordinates of the top left corner of the grid are randomly perturbed to prevent training with fixed tiles and act as data augmentation. It should be noted that this tiling procedure is only performed during training with the persistence loss and is not performed during inference.

The model is trained with the combined loss L (Eq. (3)). During the first few epochs the likelihood map is random and is not topologically informative. Thus, in the beginning of training we use DICE loss only ($\lambda = 0$). When the model starts to converge to reasonable likelihood maps, we add the persistence loss with $\lambda = 1.0$. Fig. 4 shows qualitative results. More sample results are in Appendix C.

	ShanghaiTech A		UCF QNRF	
	$\sigma = 20$ mAP/mAR	$\sigma = 5$ mAP/mAR	$\sigma = 20$ mAP/mAR	$\sigma = 5$ mAP/mAR
RAZ_Loc	58.4/74.1	19.7/42.2	28.4/48.3	3.7/14.8
TopoCount	**85.0/82.8**	**56.0/54.8**	**69.0/66.5**	**27.1/26.2**

Table 5: Localization accuracy using metric in (Liu, Weng, and Mu 2019).

3.1 Localization Performance

We evaluate *TopoCount* on several datasets using (1) localized counting; (2) F1-score matching accuracy; and (3) the NWPU localization challenge metric.

Localized Counting. We evaluate the counting performance within small grid cells and aggregate the error. The Grid Average Mean absolute Errors (GAME) metric (Guerrero-Gómez-Olmedo et al. 2015), $G(L)$, divides the image into 4^L non-overlapping cells. In Table 1, the cell count in the localization-based methods LSC-CNN (Babu Sam et al. 2020) and TopoCount is the sum of predicted dots in the cell. On the other hand, the cell count in the density map estimation methods CSRNet (Li, Zhang, and Chen 2018) and Bayesian (Ma et al. 2019) is the integral of the density map over the cell area. TopoCount achieves the lowest error especially at the finest scale (level L=3), which indicates higher localization accuracy by the predicted dots.

	Shanghai. A		Shanghai. B		UCF CC 50		UCF QNRF		JHU++		NWPU	
Model	MAE	RMSE	MAE	RMSE	MAE	RMSE	MAE	RMSE	MAE	RMSE	MAE	RMSE
IC-CNN (Ranjan, Le, and Hoai 2018)	68.5	116.2	10.7	16	260.9	365.5	-	-	-	-	-	-
CSRNet (Li, Zhang, and Chen 2018)	68.2	115	10.6	16	266.1	397.5	-	-	85.9	309.2	121.3	387.8
SANet (Cao et al. 2018)	67	104.5	8.4	13.6	258.4	334.9	-	-	91.1	320.4	190.6	491.4
ANF (Zhang et al. 2019)	63.9	99.4	8.3	13.2	250.2	340	110	174	-	-	-	-
RAZ-Net (Liu, Weng, and Mu 2019)	65.1	106.7	8.4	14.1	-	-	116	195	-	-	151.5	634.7
LSC-CNN (Babu Sam et al. 2020)	66.4	117.0	8.1	12.7	225.6	302.7	120.5	218.2	112.7	454.4	-	-
CAN (Liu, Salzmann, and Fua 2019b)	62.3	100	7.8	12.2	212.2	**243.7**	107	183	100.1	314.0	106.3	**386.5**
Bayesian (Ma et al. 2019)	62.8	101.8	7.7	12.7	229.3	308.2	**89**	**155**	75.0	299.9	**105.4**	454.2
CG-DRC (Sindagi, Yasarla, and Patel 2020)	60.2	94.0	**7.5**	**12.1**	-	-	95.5	164.3	71.0	278.6	-	-
ASNet (Jiang et al. 2020)	**57.8**	**90.1**	-	-	**174.8**	251.6	91.6	159.7	-	-	-	-
TopoCount (Ours)	61.2	104.6	7.8	13.7	184.1	258.3	**89**	159	**60.9**	**267.4**	107.8	438.5

Table 6: Counting Performance Evaluation

	ShanghaiTech A		ShanghaiTech B		UCF CC 50		UCF QNRF		JHU++	
Model	MAE	RMSE	MAE	RMSE	MAE	RMSE	MAE	RMSE	MAE	RMSE
CAN (Liu, Salzmann, and Fua 2019b)	62.3	100	7.8	**12.2**	212.2	**243.7**	107	183	100.1	314.0
TopoCount + CAN (Ours)	**59.5**	**93.3**	**7.5**	13.2	**190**	249	**99**	**162**	**71.9**	**260.9**
	-4.5%	***-6.7%***	***-2.5%***	*+8.1%*	***-10.5%***	*+2.2%*	***-7.5%***	***-11.5%***	***-28.2%***	***-19.9%***
Bayesian (Ma et al. 2019)	62.8	101.8	7.7	12.7	229.3	308.2	89	155	75.0	299.9
TopoCount + Bayesian (Ours)	**58**	**96.3**	**7.2**	**11.8**	**191**	**257**	**85**	**148**	**61.8**	**262.0**
	-7.6%	***-5.4%***	***-6.5%***	***-7.1%***	***-16.7%***	***-16.6%***	***-4.5%***	***-4.5%***	***-17.6%***	***-12.7%***

Table 7: Integration of TopoCount with density estimation methods.

Matching Accuracy. We evaluate matching accuracy in two ways. First, similar to (Idrees et al. 2018) on the UCF QNRF dataset, we perform a greedy matching between detected locations and ground truth dots at thresholds varying from 1 to 100 and average the precision, recall, and F-scores over all thresholds. We compare with scores reported in (Idrees et al. 2018) in addition to calculated scores for SOTA localization methods (Babu Sam et al. 2020). Table 2 shows that our method achieves the highest F-score.

Second, similar to (Liu, Weng, and Mu 2019) on ShanghaiTech Part A and UCF QNRF datasets, we impose at each dot annotation an un-normalized Gaussian function parameterized by σ. A true positive is a predicted dot whose response to the Gaussian function is greater than a threshold t. We compare with (Liu, Weng, and Mu 2019) results at $\sigma = 5$ and $\sigma = 20$. Table 5 reports the mean average precision (mAP) and mean average recall (mAR) for $t \in [0.5, 0.95]$, with a step of 0.05. TopoCount achieves the highest scores with a large margin at both the small and large sigma σ.

NWPU-Crowd Online Localization Challenge. NWPU dataset provides dot annotation in addition to box coordinates with specified width, w, and height, h, surrounding each head. The online challenge evaluates the F-score with 2 adaptive matching distance thresholds: $\sigma_l = \sqrt{w^2 + h^2}/2$ and a more strict threshold $\sigma_s = min(w, h)/2$. Table 3 shows the F-score, precision, and recall with the 2 thresholds against the published challenge leaderboard. TopoCount achieves the highest F-score in both thresholds. More results are in Appendix C.5.

Ablation Study for the Loss Function. On the ShanghaiTech Part A dataset, we compare the performance of TopoCount trained with variations of the loss function in Eq. 3: (1) per-pixel weighted Binary Cross Entropy (BCE) loss as in (Liu, Weng, and Mu 2019) with empirically chosen weight of 5 to account for the amount of class imbalance in the ground truth dot maps, (2) per-pixel DICE loss only (i.e $\lambda = 0$ in Eq. 3), and (3) a combined per-pixel DICE loss and Persistence loss with $\lambda \in \{0.5, 1, 1.5, 2\}$. The results in Table 4 show the training with BCE loss gives the largest error. With $\lambda = 0$, i.e., DICE without the persistence loss, the error is lower. The error is further lowered with the introduction of the persistence loss. Varying λ between 0.5 and 2.0 the results are more robust and comparable, with the best performance at $\lambda = 1$ and $\lambda = 1.5$. Consequently, we use $\lambda = 1$ in all our experiments.

3.2 Counting Performance

Our localization method can be directly applied to crowd counting task. It performs competitively among SOTA counting methods. In Table 6, we compare TopoCount's overall count in terms of the Mean Absolute Error (MAE) and Root Mean Squared Error (RMSE) against SOTA counting methods. Our method achieves SOTA performance on the new JHU++ large scale dataset and is mostly between second and third place for the other datasets compared to SOTA density-based methods. More detailed results on the JHU++ are in Appendix C.6.

Integration with Density-Based Methods. A high quality localization model can be combined with density-based methods to boost their performance. As described in Section 2.4, we integrate TopoCount with two SOTA density-based counting algorithms and report the results. Table 7

shows that the integration results in a significant improvement over the individual performance of the density map models. This further demonstrates the high quality of TopoCount localization and suggests that more sophisticated density map models can benefit from high quality localization maps to achieve an even better counting performance.

4 Conclusion

This paper proposes a novel method for localization in the crowd. We propose a topological constraint and a novel persistence loss based on persistent homology theory. The proposed topological constraint is flexible and suitable for both sparse and dense regions. The proposed method achieves state-of-the-art localization accuracy. The high quality of our results is further demonstrated by the significant boost of the performance of density-based counting algorithms when using our results as additional input. Our method closes the gap between the performance of localization and density map estimation methods; thus paving the way for advanced spatial analysis of crowded scenes in the future.

References

Aukerman, A.; Carrière, M.; Chen, C.; Gardner, K.; Rabadán, R.; and Vanguri, R. 2020. Persistent Homology Based Characterization of the Breast Cancer Immune Microenvironment: A Feasibility Study. In *36th International Symposium on Computational Geometry (SoCG))*.

Babu Sam, D.; Peri, S. V.; Narayanan Sundararaman, M.; Kamath, A.; and Radhakrishnan, V. B. 2020. Locate, Size and Count: Accurately Resolving People in Dense Crowds via Detection. *IEEE Transactions on Pattern Analysis and Machine Intelligence* .

Barua, S.; Fang, P.; Sharma, A.; Fujimoto, J.; Wistuba, I.; Rao, A. U. K.; and Lin, S. H. 2018. Spatial interaction of tumor cells and regulatory T cells correlates with survival in non-small cell lung cancer. *Lung Cancer* 117: 73–79.

Cao, X.; Wang, Z.; Zhao, Y.; and Su, F. 2018. Scale Aggregation Network for Accurate and Efficient Crowd Counting. In Ferrari, V.; Hebert, M.; Sminchisescu, C.; and Weiss, Y., eds., *Computer Vision – ECCV 2018*.

Carriere, M.; Cuturi, M.; and Oudot, S. 2017. Sliced wasserstein kernel for persistence diagrams. In *Proceedings of the 34th International Conference on Machine Learning-Volume 70*, 664–673. JMLR. org.

Chan, H. L.; Yan, S.; Lui, L. M.; and Tai, X.-C. 2017. Topology-Preserving Image Segmentation by Beltrami Representation of Shapes. *Journal of Mathematical Imaging and Vision* .

Chazal, F.; Guibas, L. J.; Oudot, S. Y.; and Skraba, P. 2013. Persistence-based clustering in riemannian manifolds. *Journal of the ACM (JACM)* 60(6): 41.

Chen, C.; Ni, X.; Bai, Q.; and Wang, Y. 2019. A Topological Regularizer for Classifiers via Persistent Homology. In *The 22nd International Conference on Artificial Intelligence and Statistics*.

Clough, J.; Oksuz, I.; Byrne, N.; Schnabel, J.; and King, A. 2019. Explicit Topological Priors for Deep-Learning Based Image Segmentation Using Persistent Homology. In *Information Processing in Medical Imaging, IPMI 2019, Proceedings*.

Edelsbrunner, H.; and Harer, J. L. 2010. *Computational topology: an introduction*. American Mathematical Soc.

Edelsbrunner, H.; Letscher, D.; and Zomorodian, A. 2000. Topological persistence and simplification. In *Proceedings 41st Annual Symposium on Foundations of Computer Science*, 454–463. IEEE.

Elphick, C. S. 2008. How you count counts: the importance of methods research in applied ecology. *Journal of Applied Ecology* 45(5): 1313–1320. doi:10.1111/j.1365-2664.2008.01545.x.

Gao, J.; Han, T.; Wang, Q.; and Yuan, Y. 2019. Domain-adaptive Crowd Counting via Inter-domain Features Segregation and Gaussian-prior Reconstruction.

Ge, W.; and Collins, R. T. 2009. Marked point processes for crowd counting. In *2009 IEEE Conference on Computer Vision and Pattern Recognition*.

Guerrero-Gómez-Olmedo, R.; Torre-Jiménez, B.; López-Sastre, R.; Maldonado-Bascón, S.; and Oñoro-Rubio, D. 2015. Extremely Overlapping Vehicle Counting. In *Pattern Recognition and Image Analysis (IbPria)*.

Hofer, C.; Kwitt, R.; Dixit, M.; and Niethammer, M. 2019. Connectivity-optimized representation learning via persistent homology. *arXiv preprint arXiv:1906.09003* .

Hofer, C.; Kwitt, R.; Niethammer, M.; and Uhl, A. 2017. Deep learning with topological signatures. In *Advances in Neural Information Processing Systems*.

Hu, P.; and Ramanan, D. 2017. Finding Tiny Faces. In *2017 IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*.

Hu, X.; Li, F.; Samaras, D.; and Chen, C. 2019. Topology-Preserving Deep Image Segmentation. In *Advances in Neural Information Processing Systems 32*, 5657–5668.

Idrees, H.; Saleemi, I.; Seibert, C.; and Shah, M. 2013. Multi-source Multi-scale Counting in Extremely Dense Crowd Images. In *IEEE Conference on Computer Vision and Pattern Recognition*.

Idrees, H.; Tayyab, M.; Athrey, K.; Zhang, D.; Al-Máadeed, S.; Rajpoot, N. M.; and Shah, M. 2018. Composition Loss for Counting, Density Map Estimation and Localization in Dense Crowds. In *ECCV*.

Jiang, X.; Zhang, L.; Xu, M.; Zhang, T.; Lv, P.; Zhou, B.; Yang, X.; and Pang, Y. 2020. Attention Scaling for Crowd Counting. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR)*.

Laradji, I. H.; Rostamzadeh, N.; Pinheiro, P. O.; Vázquez, D.; and Schmidt, M. 2018. Where Are the Blobs: Counting by Localization with Point Supervision. In *Computer Vision - ECCV - 15th European Conference*.

Lempitsky, V.; and Zisserman, A. 2010. Learning To Count Objects in Images. In *Proceedings of the 23rd International Conference on Neural Information Processing Systems*.

Li, Y.; Zhang, X.; and Chen, D. 2018. CSRNet: Dilated Convolutional Neural Networks for Understanding the Highly Congested Scenes. *IEEE Conference on Computer Vision and Pattern Recognition* .

Liu, C.; Weng, X.; and Mu, Y. 2019. Recurrent Attentive Zooming for Joint Crowd Counting and Precise Localization. In *IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR)*.

Liu, J.; Gao, C.; Meng, D.; and Hauptmann, A. G. 2018. DecideNet: Counting Varying Density Crowds Through Attention Guided Detection and Density Estimation. In *IEEE Conference on Computer Vision and Pattern Recognition,*.

Liu, W.; Salzmann, M.; and Fua, P. 2019a. Context-Aware Crowd Counting. *IEEE Conference on Computer Vision and Pattern Recognition (CVPR)* .

Liu, W.; Salzmann, M.; and Fua, P. 2019b. Context-Aware Crowd Counting. In *The IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*.

Ma, Z.; Lei Yu; and Chan, A. B. 2015. Small instance detection by integer programming on object density maps. In *IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*.

Ma, Z.; Wei, X.; Hong, X.; and Gong, Y. 2019. Bayesian Loss for Crowd Count Estimation With Point Supervision. In *The IEEE International Conference on Computer Vision (ICCV)*.

Mcphail, C.; and McCarthy, J. 2004. Who Counts and How: Estimating the Size of Protests. *Contexts* 3: 12–18. doi: 10.1525/ctx.2004.3.3.12.

Mosinska, A.; Márquez-Neila, P.; Kozinski, M.; and Fua, P. 2018. Beyond the Pixel-Wise Loss for Topology-Aware Delineation. In *IEEE Conference on Computer Vision and Pattern Recognition, CVPR*.

Munkres, J. R. 2018. *Elements of algebraic topology*. CRC Press.

Ni, X.; Quadrianto, N.; Wang, Y.; and Chen, C. 2017. Composing tree graphical models with persistent homology features for clustering mixed-type data. In *Proceedings of the 34th International Conference on Machine Learning-Volume 70*, 2622–2631. JMLR. org.

Poulenard, A.; Skraba, P.; and Ovsjanikov, M. 2018. Topological function optimization for continuous shape matching. In *Computer Graphics Forum*, volume 37.

Ranjan, V.; Le, H.; and Hoai, M. 2018. Iterative Crowd Counting. In *ECCV*.

Reininghaus, J.; Huber, S.; Bauer, U.; and Kwitt, R. 2015. A stable multi-scale kernel for topological machine learning. In *Proceedings of the IEEE conference on computer vision and pattern recognition*, 4741–4748.

Ren, S.; He, K.; Girshick, R.; and Sun, J. 2015. Faster R-CNN: Towards Real-Time Object Detection with Region Proposal Networks. In *Advances in Neural Information Processing Systems 28*.

Ren, W.; Kang, D.; Tang, Y.; and Chan, A. B. 2018. Fusing Crowd Density Maps and Visual Object Trackers for People Tracking in Crowd Scenes. In *IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*.

Ronneberger, O.; P.Fischer; and Brox, T. 2015. U-Net: Convolutional Networks for Biomedical Image Segmentation. In *Medical Image Computing and Computer-Assisted Intervention (MICCAI)*.

Simonyan, K.; and Zisserman, A. 2015. Very Deep Convolutional Networks for Large-Scale Image Recognition. In *International Conference on Learning Representations*.

Sindagi, V. A.; Yasarla, R.; and Patel, V. M. 2020. JHU-CROWD++: Large-Scale Crowd Counting Dataset and A Benchmark Method. *Technical Report* .

Sudre, C. H.; Li, W.; Vercauteren, T.; Ourselin, S.; and Jorge Cardoso, M. 2017. Generalised Dice Overlap as a Deep Learning Loss Function for Highly Unbalanced Segmentations. In *Deep Learning in Medical Image Analysis and Multimodal Learning for Clinical Decision Support*.

Waggoner, J.; Zhou, Y.; Simmons, J.; Graef, M. D.; and Wang, S. 2015. Topology-Preserving Multi-label Image Segmentation. In *2015 IEEE Winter Conference on Applications of Computer Vision*, 1084–1091.

Wang, Q.; Gao, J.; Lin, W.; and Li, X. 2020. NWPU-Crowd: A Large-Scale Benchmark for Crowd Counting and Localization. *IEEE Transactions on Pattern Analysis and Machine Intelligence* .

Wang, Q.; Gao, J.; Lin, W.; and Yuan, Y. 2019. Learning From Synthetic Data for Crowd Counting in the Wild. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR)*.

Yang, D.; Yurtsever, E.; Renganathan, V.; Redmill, K.; and Özgüner, U. 2020. A vision-based social distancing and critical density detection system for COVID-19. *Image video Process. DOI* .

Zhang, A.; Yue, L.; Shen, J.; Zhu, F.; Zhen, X.; Cao, X.; and Shao, L. 2019. Attentional Neural Fields for Crowd Counting. In *The IEEE International Conference on Computer Vision (ICCV)*.

Zhang, Y.; Zhou, D.; Chen, S.; Gao, S.; and Ma, Y. 2016. Single-Image Crowd Counting via Multi-Column Convolutional Neural Network. In *The IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*.

Zhao, Q.; and Wang, Y. 2019. Learning metrics for persistence-based summaries and applications for graph classification. In *Advances in Neural Information Processing Systems*.

Zhao, T.; Nevatia, R.; and Wu, B. 2008. Segmentation and Tracking of Multiple Humans in Crowded Environments. *IEEE Transactions on Pattern Analysis and Machine Intelligence* .

A Persistent Homology Computation

Algorithm 1: Computing Persistence

Data: Likelihood map f
Result: Paired modes and saddles, $\mathcal{P} = \{(m_i, s_i)\}$.

Build a grid graph $G = (V, E)$. Nodes are all pixels. Edges connect adjacent nodes;
Sort nodes in decreasing order of their likelihood f, $\widehat{V} = (v_1, v_2, \ldots)$, $f(v_i) \geq f(v_{i+1})$;
Initialize the visited list visited(u) = false $\forall u \in V$;
Initialize a component list, $\mathcal{C} = \emptyset$;
Initialize the persistence pair list, $\mathcal{P} = \emptyset$;
forall $u \in \widehat{V}$ **do**
  $\mathcal{N}_u = \{\text{neighbors of } u\}$;
  Components adjacent to u: $\mathcal{C}_u = \emptyset$;
  /* Find all neighbor components of u. */
  forall $v \in \mathcal{N}_u$ **do**
    if visited(v) **then**
      $C_v =$ the component in $\mathcal{C}$ such that $v \in C_v$;
      $\mathcal{C}_u = \mathcal{C}_u \cup \{C_v\}$
  /* If u has no neighbor components, it will create a new component. */
  if $|\mathcal{C}_u| == 0$ **then**
    $C = \{u\}$ $\mathcal{C} = \mathcal{C} \cup \{C\}$;
  /* If u has a single neighbor component, it will merge with it. */
  else if $|\mathcal{C}_u| == 1$ **then**
    $C =$ the only component in $\mathcal{C}_u$;
    $C = C \cup \{u\}$;
  /* Case u has multiple neighbor components: (1) Find neighbor component with earliest birth C_{max}. (2) u becomes saddle point for all other neighbor components. (3) Merge all other neighbor components with C_{max}. */
  else
    $C_{max} = \text{argmax}_{C \in \mathcal{C}_u} \text{birth}(C)$;
    $\mathcal{C} = (\mathcal{C} \backslash \mathcal{C}_u) \cup \{C_{max}\}$;
    forall $C \in \mathcal{C}_u \backslash \{C_{max}\}$ **do**
      $m = \text{argmax}_{w \in C} f(w)$;
      $s = u$;
      $\mathcal{P} = \mathcal{P} \cup \{(m, s)\}$;
      $C_{max} = C_{max} \cup C$;
  visited(u) = true;
return $\mathcal{P}$;

Persistent homology. The theory of persistent homology (Edelsbrunner, Letscher, and Zomorodian 2000; Edelsbrunner and Harer 2010) measures the saliency of different topological structures from a given scalar function (the likelihood function f in our setting). We threshold an image patch, δ, at a given level t and denote by $\delta_f^t = \{x \in \delta | f(x) \geq t\}$ the *superlevel set*. A superlevel set can have topological structures of different dimensions. For example: 0-dimensional structures are connected components and 1-dimensional structures are handles/holes. [1] We focus on 0-dimensional topology in this paper, although the theory covers topology of all dimensions. The theory of persistent homology tracks the life span of all topological structures of the superlevel set as we continuously change the threshold t.

We continuously decrease t from $+\infty$ to $-\infty$. As t decreases, we track the connected components of the progressively growing superlevel set, δ_f^t. During the process, a mode (i.e., local maximum) gives birth to a new connected component. The component dies when it touches another component created by a higher mode. The location at which the two components meet is a saddle point, s. The function values of the mode and the saddle point are called the *birth* and *death times*. We use their difference, called the *persistence*, to measure the saliency of this mode. See Fig. 2(g) in the paper for an illustration. A pseudo code of the algorithm to find the 0-dimensional components and their persistence is outlined in Algorithm 1.

B Additional Training and Implementation Details

Model Architecture We use a U-Net style architecture. The detailed per-layer architecture is shown in Fig. 5.

Encoder

Layer Type	Output Channels	Feed Direction
Conv2D (3x3)	64	↓
ReLU	64	↓
Conv2D (3x3)	64	↓
ReLU	64	↓
MaxPool2D	64	↓
Conv2D (3x3)	128	↓
ReLU	128	↓
Conv2D (3x3)	128	↓
ReLU	128	↓
MaxPool2D	128	↓
Conv2D (3x3)	256	↓
ReLU	256	↓
Conv2D (3x3)	256	↓
ReLU	256	↓
Conv2D (3x3)	256	↓
ReLU	256	↓
MaxPool2D	256	↓
Conv2D (3x3)	512	↓
ReLU	512	↓
Conv2D (3x3)	512	↓
ReLU	512	↓
Conv2D (3x3)	512	↓
ReLU	512	↓
MaxPool2D	512	↓
Conv2D (3x3)	512	↓
ReLU	512	↓
Conv2D (3x3)	512	↓
ReLU	512	↓
Conv2D (3x3)	512	↓
ReLU	512	↓

Decoder

Layer Type	Output Channels	Feed Direction
Conv2D (1x1)	1	
ReLU	64	↑
Conv2D (3x3)	64	↑
ReLU	64	↑
Conv2D (3x3)	64	↑
ConvTranspose2d	64	↑
ReLU	128	↑
Conv2D (3x3)	128	↑
ReLU	128	↑
Conv2D (3x3)	128	↑
ConvTranspose2d	128	↑
ReLU	256	↑
Conv2D (3x3)	256	↑
ReLU	256	↑
Conv2D (3x3)	256	↑
ConvTranspose2d	256	↑
ReLU	512	↑
Conv2D (3x3)	512	↑
ReLU	512	↑
Conv2D (3x3)	512	↑
ConvTranspose2d	512	↑

Figure 5: TopoCount detailed model architecture.

Image Scaling. For all crowd counting datasets we use the original images without scaling, except for the UCF QNRF. For UCF QNRF dataset we apply the following policy: during training the images are resized so that the longer side is of maximum length 2048, and during test we resize so that

[1] For a more rigorous definition, please refer to classic algebraic topology textbooks (Munkres 2018).

the shorted side is of maximum length 2048. This more relaxed resizing policy during test allows the model to capture more details in the densely crowded regions of the test images. On the other hand, the variation in scale is not an issue in the JHU++ and NYPU-Crowd datasets because we have extra information of the head box size and the dot dilation size is selected proportionally.

Training Crop size. For the datasets that have an average resolution less than 1024×1024, i.e. ShanghaiTech Parts A and B, we use the whole image during training. For the rest of the datasets we train with crops of size min(image width,1024) $\times$ min(image height,1024).

Training Optimizer. We train TopoCount with Adam optimizer at a learning rate of 0.00005, using a batch size of 1.

Postprocessing. The model generates a likelihood of the topology map. The final mask of the topology map is obtained by thresholding the likelihood. We empirically choose a double thresholding procedure with high threshold = 0.5, and low threshold = 0.4. In particular, the high threshold is used to first filter the domain and select the connected components representing each person. Next, we lower the threshold just to grow the selected connected components. This is to get the right geometry so that the center of each connected component is closer to the corresponding true dot. The dots are estimated as the centers of the connected components in the mask. The resulting dots are used as the final output and are used in our evaluations with various metrics.

Software and Hardware The implementation used Python 3.6 and Pytorch version 0.5.0a0. The models were training on system with Ubuntu operating system, and NVidia Volta GPU. The amount of GPU memory utilized varies by batch size and crop size. For our configurations it used up less than 10 GB GPU memory.

C Additional Experimental Results

C.1 Additional Qualitative Results

Fig. 6 shows additional qualitative results. It shows samples of the topology and density maps estimated by TopoCount and by Bayesian (Ma et al. 2019) + TopoCount, respectively. The F-scores and counting errors are reported next to the figures. The topological map closely matches the ground truth dots annotation arrangement; as does the density map. Additionally, Fig. 8 and Fig. 9 show sample results of TopoCount on some difficult cases from the JHU++ dataset. Note that the F-scores reported in all the qualitative results in both the paper and the supplementary material represent the mean of the F-scores calculated using matching distance thresholds ranging from 1 to 100, as proposed in (Idrees et al. 2018).

C.2 Integration with Density Map Results

Fig. 7 shows samples from the density map estimation by the integration of TopoCount and the baseline density map-based method (Ma et al. 2019). We see in the the figure how the additional information provided by the topological map improves the quality of the density map estimation. In the first sample, the density map of the baseline is blurry while baseline+TopoCount gives a more structured density map that is closer to the ground truth density map. In the second sample, the region indicated by the red ellipse is a densely crowded region that is in the shadow. It is missed by the baseline while TopoCount's topology map is able to identify the crowd. By the integration in baseline+TopoCount, this extra information is passed on to baseline+TopoCount and it also recovers the crowd in the shadow.

C.3 Ablation Study: Choosing Patch Size Selection for Persistence Loss

The window size of the topological constraint patch controls the level of localization we would like to focus on. In the one extreme, when the patch is 1×1, the topological constraint becomes a per-pixel supervision. It helps the model to learn features for dot pixels, but loses the rich topological information within local neighborhoods. It is also not flexible/robust with perturbation. On the other extreme, when the patch is the whole image, the topological information is simply the total count of the image. This information is too high-level and will not help the model to learn efficiently; thus we have all other intermediate level supervisions, such as the density map. A properly chosen patch size will exploit rich spatial relationships within local neighborhoods while being robust to perturbation. In our experiments, we use a patch size of 50×50 pixels for ShanghaiTech and UCF CC 50 datasets, for datasets with larger variation in scale, namely UCF QNRF, JHU++, and NWPU-Crowd datasets, we use a larger patch size of 100×100 pixels. Next we explain how we choose the patch sizes.

To select the patch size for the persistence loss, we train 4 models on the ShanghaiTech Part A dataset with different patch sizes: 150×150 , 100×100, 50×50, and 30×30. We evaluate the models localization accuracy using the GAME metric at different scales L = 1 through 3, see Table 8. Training with patch size 30 or 150 yields poor performance. On the other hand, training with patch sizes 50 or 100 gives mostly similar results except at the smallest cell size (L=3) where patch size 50 is the winner, indicating better localization. We thus choose patch size of 50 for the ShanghaiTech and UCF CC 50 experiments.

The datasets UCF QNRF, JHU++, and NWPU-Crowd are different from the other datasets in their wide variation in scale and resolution. We suspect that a patch size of 50 may not be suitable. We experiment with a small subset of randomly selected (N=50) images from the UCF QNRF training data. Again, we train 4 models with different patch sizes: 150, 100, 50, and 30, and evaluate the models localization using GAME. Because the images in this dataset have a higher resolution range, we use L = 1 through 4, see Table 9. We find that a patch size of 150 is more suitable at the coarser cells (L=1, 2) while a patch size of 50 is more suitable at the finer cells (L=3, 4). Therefore for training on the aforementioned datasets we choose the intermediate patch size of 100.

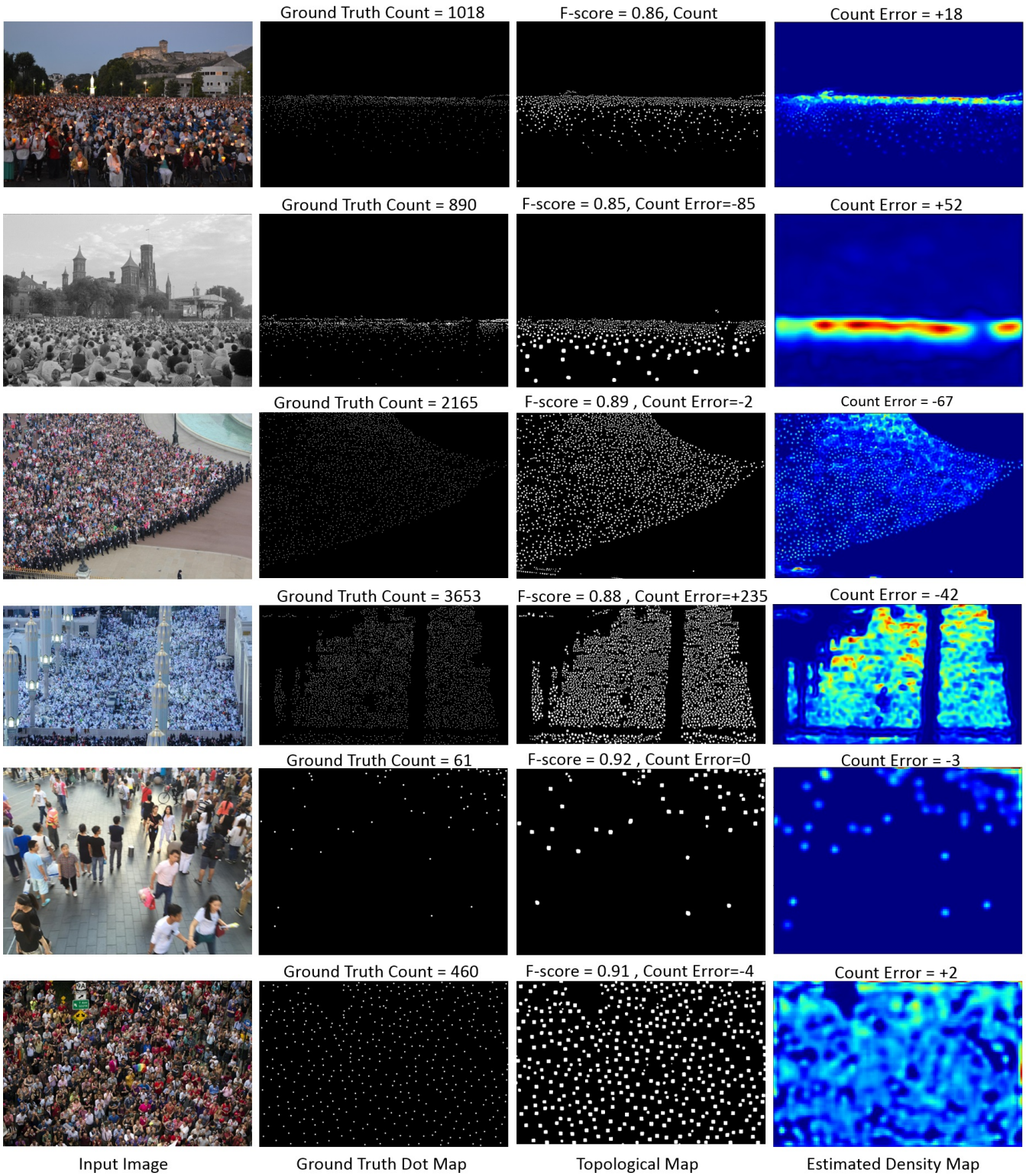


Figure 6: Sample results from different density crowd images. The columns represent the original image, ground truth and topological maps by TopoCount, and the estimated density map by the integration of Bayesian (Ma et al. 2019) + TopoCount.

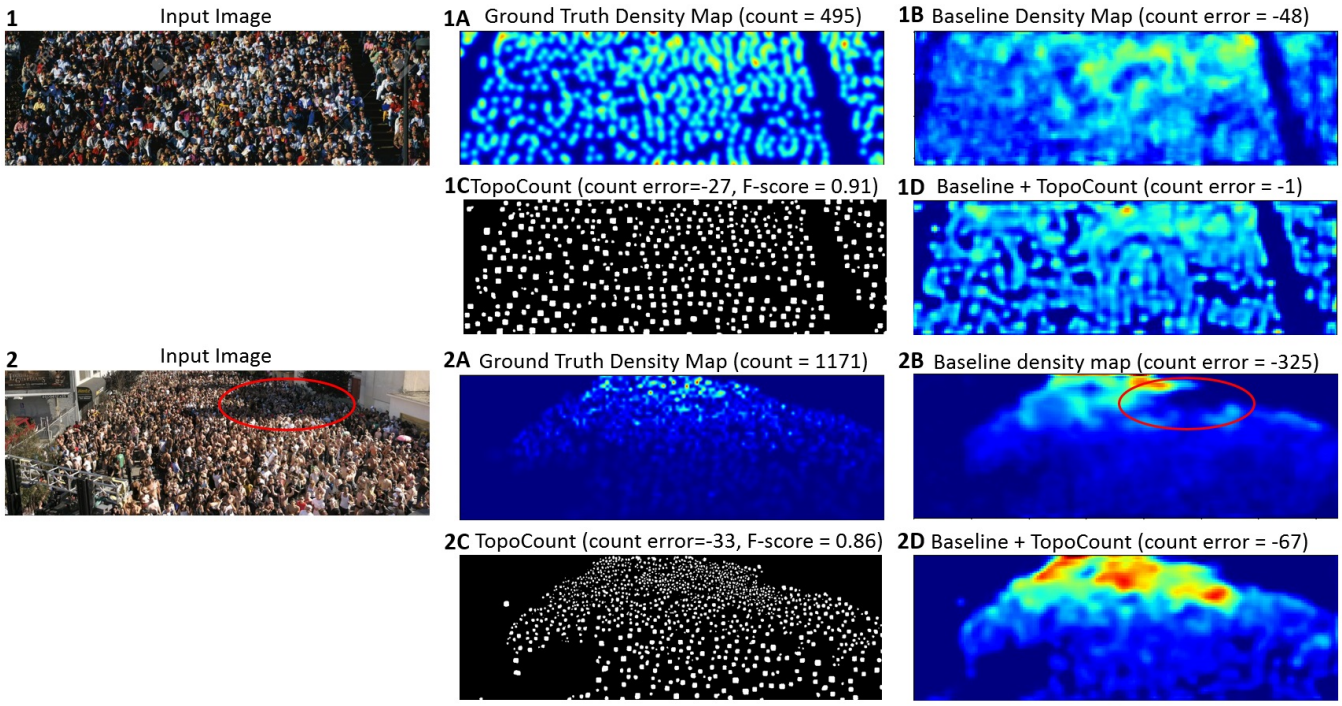


Figure 7: Sample results from crowd counting datasets. For each sample we show: the original image, the ground truth density map (A), the baseline density map (by Bayesian loss (Ma et al. 2019)) (B), the topological map by TopoCount (C), and the density map by baseline + TopoCount (D). With the addition of the topological map, the estimated density map (D) has better topological structure and fixes shadowed regions missed by the baseline (Ma et al. 2019) (B).

Patch Size	G(L)		
	G(1)	G(2)	G(3)
150	75.4	89.9	114.2
100	**68.4**	82.0	107.7
50	69.3	**81.6**	**104.9**
30	75.4	86.4	108.2

Table 8: Ablation study on the patch size for persistence loss training on ShanghaiTech Part A dataset. Evaluate localization score GAME(L) with patch sizes 150, 100, 50, and 30.

Patch Size	G(L)			
	G(1)	G(2)	G(3)	G(4)
150	**153.5**	**175.1**	208.5	272.3
100	155.9	177.1	**206.6**	264.4
50	160.6	179.4	206.8	**260.3**
30	179.9	195.2	222.1	273.1

Table 9: Ablation study on the patch size for persistence loss training on UCF QNRF (N=50). Evaluate localization score GAME(L) with patch sizes 150, 100, 50, and 30.

C.4 Shanghai Part A and UCF QNRF Localization Accuracy

We present more detailed results from the evaluation of the matching accuracy reported in Table 5 of the paper. We are using the matching metric proposed in (Liu, Weng, and Mu 2019). At each dot annotation we impose an un-normalized Gaussian function parameterized by σ. A true positive is a predicted dot whose response to the Gaussian function is greater than a threshold t. We compare against (Liu, Weng, and Mu 2019) at $\sigma \in \{40, 20, 5\}$. The smaller the value of σ the closer the prediction needs to be to the ground truth dot to be counted as a true positive. Table 10 reports $AP.5$ and $AR.5$: the average precision and recall at $t = 0.5$, and $AP.75$ and $AR.75$: the average precision and recall at $t = 0.75$, in addition to the mean average precision (mAP) and mean average recall (mAR) for $t \in [0.5, 0.95]$, with a step of 0.05. TopoCount achieves the highest scores with a large margin at both the small and large sigma σ.

C.5 NYPU-Crowd Online Localization Challenge

To train on the NWPU-Crowd dataset, we use all training images including those with no heads. The model is trained with crops of 1024x1024 pixels of the original image using the original image sizes. NWPU dataset provides dot annotation in addition to box coordinates with specified width, w, and height, h, surrounding each head. The online challenge evaluates the F-score with 2 adaptive matching distance thresholds: $\sigma_l = \sqrt{w^2 + h^2}/2$ and a more strict

	ShanghaiTech A								
	$\sigma = 40$			$\sigma = 20$			$\sigma = 5$		
	AP.50/AR.50	AP.75/AR.75	mAP/mAR	AP.50/AR.50	AP.75/AR.75	mAP/mAR	AP.50/AR.50	AP.75/AR.75	mAP/mAR
RAZ_Loc	74.5/84.7	69.9/82.0	69.1/81.2	66.7/79.9	60.1/75.3	58.4/74.1	36.0/40.9	20.5/**57.9**	19.7/42.2
TopoCount	**91.0**/**88.6**	**89.6**/**87.2**	**89.2**/**86.8**	**88.6**/**86.2**	**86.1**/**83.8**	**85.0**/**82.8**	**72.5**/**70.9**	**57.6**/56.4	**56.0**/**54.8**
	UCF QNRF								
RAZ_Loc	57.3/71.9	48.1/65.2	46.2/63.6	41.4/60.2	28.7/49.7	28.4/48.3	7.9/24.2	3.1/14.3	3.7/14.8
TopoCount	**87.5**/**84.2**	**83.3**/**80.2**	**81.7**/**78.6**	**79.7**/**76.7**	**71.1**/**68.6**	**69.0**/**66.5**	**41.7**/**40.3**	**26.3**/**25.3**	**27.1**/**26.2**

Table 10: Localization accuracy using metric in (Liu, Weng, and Mu 2019).

threshold $\sigma_s = min(w, h)/2$. In Table 3 of the paper we showed the F-score, precision, and recall with the 2 thresholds against the published challenge leaderboard. Here we show more detailed results. For each threshold the recall is further categorized by the head bounding box area range. In Table 11, $A0 \sim A5$ correspond to the head area ranges: $[10^0, 10^1], (10^1, 10^2], (10^2, 10^3], (10^3, 10^4], (10^4, 10^5]$, and $> 10^5$, respectively. We see that TopoCount achieves highest recall in the smaller head range catergories while TinyFaces (Hu and Ramanan 2017) achieves highest recall in the larger head range catergories.

C.6 JHU++ Counting Evaluation

The model is trained with crops of 1024x1024 pixels of the original image without any resizing. There are a few images in the training set with no heads at all. We did not use them in training. The dataset contains images with varying difficulties including weather conditions such as rain and fog. We report in Table 12 and Table 13 the categorized counting performance on the validation and test sets, respectively. The categories are: *Low*: images containing count between 0 and 50, *Medium*: images containing count between 51 and 500, *High*: images with count more than 500 people, *Weather*: weather degraded images, and *Overall*: all the images in the set. We see that TopoCount and the integration of TopoCount with density-based methods achieve the best performance across most categories. Fig. 8 and Fig. 9 show sample results of TopoCount on some difficult cases in the JHU++ dataset.

Method	σ_l		σ_s	
	A0 / A1 / A2 / A3 / A4 / A5	Average	A0 / A1 / A2 / A3 / A4 / A5	Average
Faster RCNN (Ren et al. 2015)	0.0 / 00.0 / 00.0 / 07.9 / 37.2 / 63.5	18.2	0.0 / 00.0 / 00.0 / 07.3 / 35.4 / 60.2	17.2
TinyFaces (Hu and Ramanan 2017)	4.2 / 22.6 / 59.1 / **90.0** / **93.1** / **89.6**	59.8	3.7 / 19.6 / 54.1 / **85.8** / **89.7** / **84.3**	**56.2**
VGG+GPR (Gao et al. 2019)	3.1 / 27.2 / 49.1 / 68.7 / 49.8 / 26.3	37.4	2.7 / 18.6 / 37.8 / 63.0 / 45.7 / 16.1	30.6
RAZ_Loc (Liu, Weng, and Mu 2019)	5.1 / 28.2 / 52.0 / 79.7 / 64.3 / 25.1	42.4	**4.6** / 20.7 / 43.3 / 75.2 / 60.2 / 15.9	36.7
TopoCount (ours)	**5.7** / **40.6** / **70.7** / 82.4 / 85.3 / 86.2	**61.8**	**4.6** / **28.1** / **60.3** / 79.0 / 81.2 / 77.5	55.2

Table 11: NWPU-Crowd Online Localization Challenge. The table reports the recall at different head area ranges and their average.

JHU++ Category	Low		Medium		High		Weather		Overall	
Model	MAE	RMSE	MAE	RMSE	MAE	RMSE	MAE	RMSE	MAE	RMSE
MCNN (Zhang et al. 2016)	90.6	202.9	125.3	259.5	494.9	856.0	241.1	532.2	160.6	377.7
CSRNet (Li, Zhang, and Chen 2018)	22.2	40.0	49.0	99.5	302.5	669.5	83.0	168.7	72.2	249.9
SANet (Cao et al. 2018)	13.6	26.8	50.4	78.0	397.8	749.2	72.2	126.7	82.1	272.6
SFCN (Wang et al. 2019)	11.8	19.8	39.3	73.4	297.3	679.4	**52.3**	**93.6**	62.9	247.5
LSC-CNN (Babu Sam et al. 2020)	6.8	**10.1**	39.2	64.1	504.7	860.0	77.6	187.2	87.3	309.0
JHU++ (Sindagi, Yasarla, and Patel 2020)	11.7	24.8	35.2	57.5	273.9	676.8	54.0	106.8	57.6	244.4
CAN (Liu, Salzmann, and Fua 2019b)	34.2	69.5	65.6	115.3	336.4	619.7	101.8	179.3	89.5	239.3
Bayesian (Ma et al. 2019)	6.9	10.3	39.7	85.2	279.8	620.4	58.9	124.7	59.3	229.2
CAN+TopoCount (**ours**)	26.5	49.9	38.2	63.3	277.1	621.4	62.5	112.0	64.9	227.3
Bayesian + TopoCount (**ours**)	**6.3**	11.0	32.5	58.8	**269.6**	**602.1**	62.6	123.7	**54.1**	**218.1**
TopoCount (**ours**)	6.9	11.1	**32.1**	**51.8**	275.7	633.8	57.5	119.3	54.3	228.2

Table 12: Categorical Counting Results On JHU-CROWD++ Dataset ("**Val Set**"). The descriptions of the categories are in Appendix C.6

JHU++ Category	Low		Medium		High		Weather		Overall	
Model	MAE	RMSE	MAE	RMSE	MAE	RMSE	MAE	RMSE	MAE	RMSE
MCNN (Zhang et al. 2016)	97.1	192.3	121.4	191.3	618.6	1,166.7	330.6	852.1	188.9	483.4
CSRNet (Li, Zhang, and Chen 2018)	27.1	64.9	43.9	71.2	356.2	784.4	141.4	640.1	85.9	309.2
SANet (Cao et al. 2018)	17.3	37.9	46.8	69.1	397.9	817.7	154.2	685.7	91.1	320.4
SFCN (Wang et al. 2019)	16.5	55.7	38.1	59.8	341.8	758.8	122.8	606.3	77.5	297.6
LSC-CNN (Babu Sam et al. 2020)	10.6	31.8	34.9	55.6	601.9	1,172.2	178.0	744.3	112.7	454.4
JHU++ (Sindagi, Yasarla, and Patel 2020)	14.0	42.8	35.0	53.7	314.7	712.3	**120.0**	**580.8**	71.0	278.6
CAN (Liu, Salzmann, and Fua 2019b)	37.6	78.8	56.4	86.2	384.2	789.0	155.4	617.0	100.1	314.0
Bayesian (Ma et al. 2019)	10.1	32.7	34.2	54.5	352.0	768.7	140.1	675.7	75.0	299.9
CAN+TopoCount (**ours**)	30.7	60.3	38.3	63.9	**275.0**	**659.7**	123.2	625.7	71.9	**260.9**
Bayesian+TopoCount (**ours**)	**7.8**	22.8	**28.5**	52.5	286.8	670.6	122.9	639.2	61.8	262.0
TopoCount (**ours**)	8.2	**20.5**	28.9	**50.0**	282.0	685.8	120.4	635.1	**60.9**	267.4

Table 13: Categorical Counting Results On JHU-CROWD++ Dataset ("**Test Set**"). The descriptions of the categories are in Section **??**

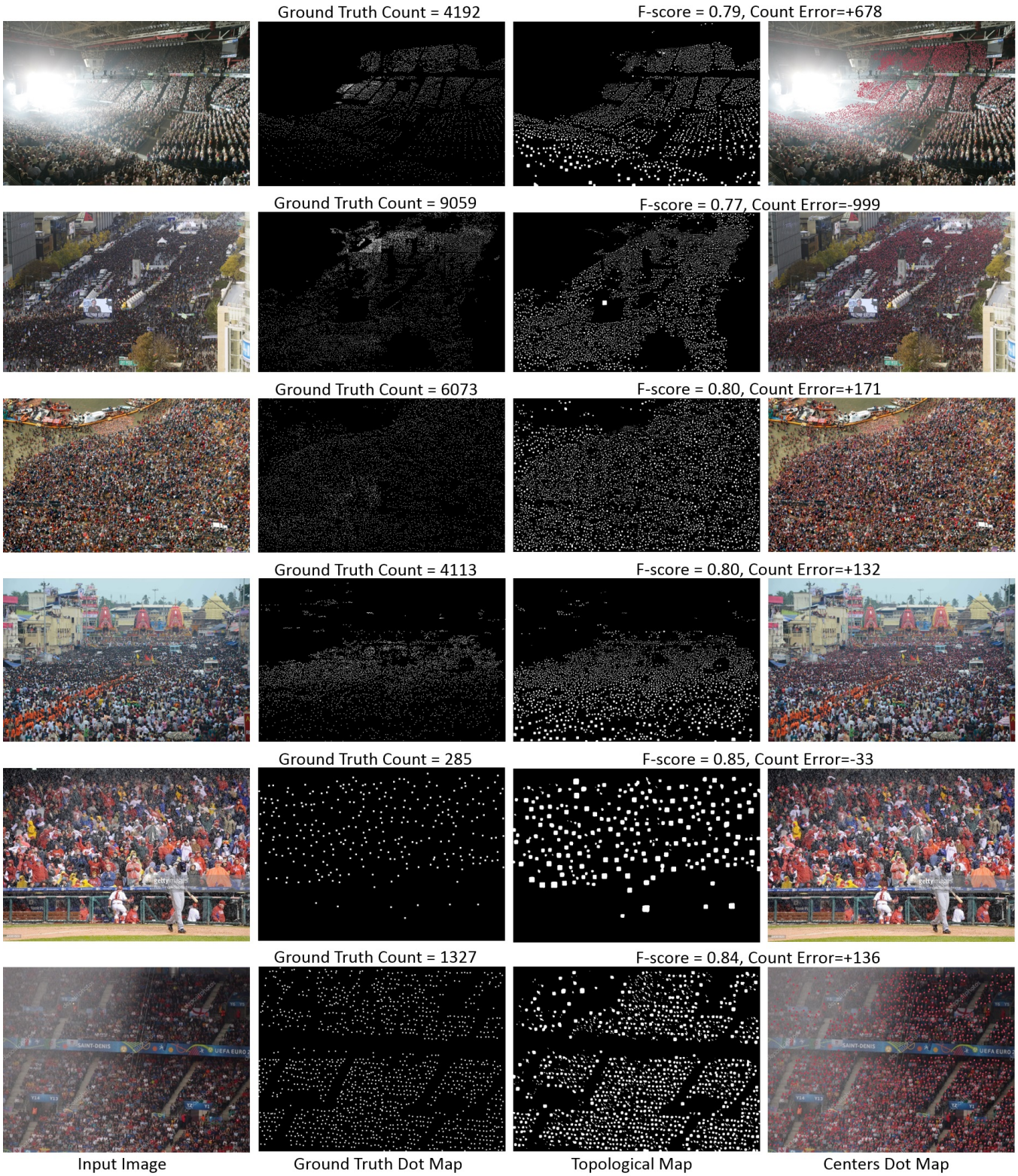


Figure 8: Sample results from some difficult cases in the JHU++ crowd counting dataset. The columns from left to right are: the input image, the ground truth dot map, the predicted topological map, and the centers of the components in the topological map overlayed on the input image as red dots.

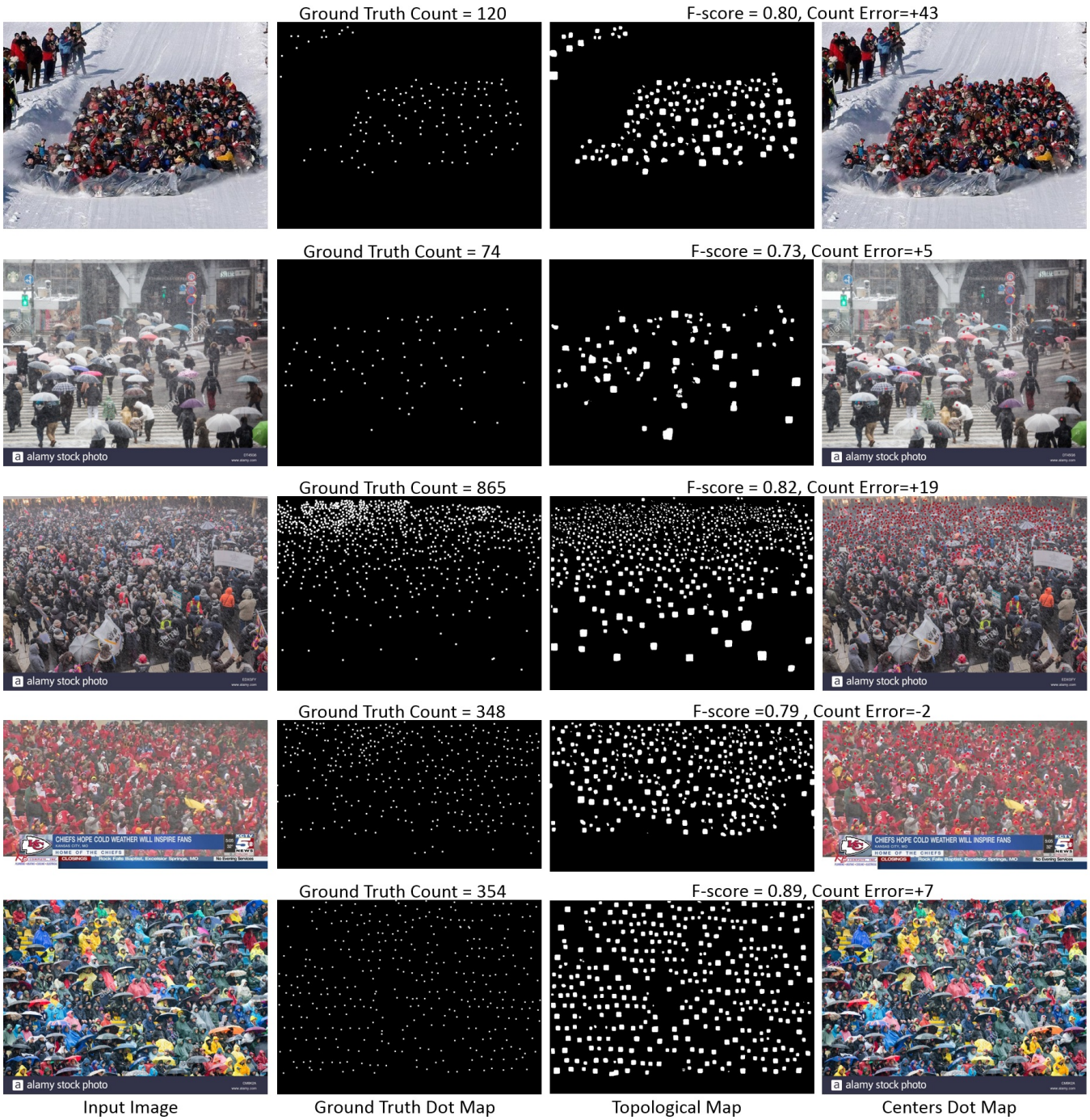


Figure 9: Sample results from some difficult cases in the JHU++ crowd counting dataset. The columns from left to right are: the input image, the ground truth dot map, the predicted topological map, and the centers of the components in the topological map overlayed on the input image as red dots.