Uncertainty-aware Continuous Implicit Neural Representations for Remote Sensing Object Counting

Siyuan Xu[1] Yucheng Wang[1] Mingzhou Fan[1] Byung-Jun Yoon[1,3] Xiaoning Qian[1,2,3]
[1]Electrical & Computer Engineering, [2]Computer Science & Engineering, Texas A&M University
[3]Computational Science Initiative (CSI), Brookhaven National Laboratory

Abstract

Many existing object counting methods rely on density map estimation (DME) of the discrete grid representation by decoding extracted image semantic features from designed convolutional neural networks (CNNs). Relying on discrete density maps not only leads to information loss dependent on the original image resolution, but also has a scalability issue when analyzing high-resolution images with cubically increasing memory complexity. Furthermore, none of the existing methods can offer reliable uncertainty quantification (UQ) for the derived count estimates. To overcome these limitations, we design UNcertainty-aware, hypernetwork-based Implicit neural representations for Counting (UNIC) to assign probabilities and the corresponding counting confidence over continuous spatial coordinates. We derive a sampling-based Bayesian counting loss function and develop the corresponding model training algorithm. UNIC outperforms existing methods on the Remote Sensing Object Counting (RSOC) dataset with reliable UQ and improved interpretability of the derived count estimates. Our code is available at https://github.com/SiyuanXu-tamu/UNIC.

1 INTRODUCTION

As urban populations have surged and urbanization has advanced swiftly, geographic entities like buildings and cars have increasingly congregated and densely populated. Consequently, this has spurred a growing research interest in scene comprehension through the lens of object counting. Object counting holds promising potential for handling similar tasks in other domains too, including crowd counting for security (Ma et al., 2019; Li et al., 2018), animal crowd estimations (Ma et al., 2015), and cell counting for biomedicine (Paul Cohen et al., 2017). People have achieved good performance by introducing deep learning (Fu et al., 2015) and self-attention (Gao et al., 2020) into counting tasks. The best-performing methods are mostly based on density map estimation (DME) (Ma et al., 2019; Gao et al., 2022). They all train convolutional neural networks (CNNs) to generate discrete density maps, which faces the following challenges: 1) The “ground-truth” density maps are always generated by convolving the ground-truth dot map with a Gaussian kernel (Wan and Chan, 2019), which may lead to biased object count estimates due to blurring effects dependent on image grid resolution as the derived density maps are not continuous (Ma et al., 2019); 2) Highly dense images often have significantly overlapping objects so that accurate estimation needs high-resolution density maps requiring additional computing resources, making it hard to handle high-resolution images; 3) There lacks reliable uncertainty quantification, which is critical for consequent decision making in many real-world applications.

To alleviate these issues, we design a new UNcertainty-aware Implicit neural representation based Counting (UNIC) method to learn a continuous function for DME, which can calculate the density distribution for any continuous spatial location in the image domain to provide fine details not limited by the grid resolution but by the capacity of the underlying network architecture. Besides, to preserve continuous information as much as possible, improve scalability, and enhance the model training convergence, we design a Bayesian loss function and derive the training algorithm for UNIC with sampling-based Bayesian counting loss estimation. To further improve convergence, we leverage hypernetworks (Skorokhodov et al., 2021)

Proceedings of the 27th International Conference on Artificial Intelligence and Statistics (AISTATS) 2024, Valencia, Spain. PMLR: Volume 238. Copyright 2024 by the author(s).

to directly learn the model parameters in UNIC, making sure that the representation function for each image is unique in an amortized fashion.

Our contributions include: 1) We design UNIC, a novel hypernetwork-based implicit neural representation (INR) for object counting, which learns a continuous function for generating density maps instead of typical discrete density maps; 2) We derive an efficient model training scheme for counting tasks based on a Bayesian counting loss estimated via sampling; 3) UNIC enables uncertainty quantification in object counting; and 4) UNIC consistently outperforms other counting methods across image resolutions in our experiments.

2 RELATED WORK

Object counting: The original ideas (Lin et al., 2001) for estimating crowd counts are based on detecting or segmenting individual objects in the scene. To sidestep the intricacies associated with the more complicated detection problem, especially with densely populated scenes, researchers have suggested directly mapping image features to object counts (Chan et al., 2008). Wang et al. (2015) adopted the AlexNet (Krizhevsky et al., 2012) architecture to predict the scalar count value as the output of the final fully connected layer.

Since introduced by Lempitsky and Zisserman (2010), counting via predicting a density map, or DME, achieves higher counting accuracy because more image information can be utilized. In this context, each pixel's value in the density map represents the estimated likelihood of an object being present in the corresponding area of the image. Subsequently, the total count of objects can be determined by integrating over this density map. CNN-based DME methods have been demonstrated to outperform conventional object counting techniques using handcrafted image features (Fu et al., 2015). The recently developed ASPDNet integrates attention and deformable convolution modules to address challenges in counting such as complex cluttered background, viewing perspective, object appearance, and size variability, with demonstrated superior counting performance on their constructed the RSOC (Remote Sensing Object Counting) dataset (Gao et al., 2020).

Implicit neural representations: Since Stanley (2007) pioneered a neuroevolution-based model augmenting extracted deep image features leveraging spatial coordinate information, Implicit Neural Representations (INRs) become well known after being applied to 3D shape representation (Mescheder et al., 2019) and other tasks, including texture analysis (Oechsle et al., 2019) and medical image analysis (Barrowclough et al., 2021). More recently, Tancik et al. (2020) and Sitzmann et al. (2020) further introduced Fourier positional encoding and periodic activation functions in INRs to learn high-frequency details and better represent complex natural signals. When combined with generative models including INRs, hypernetworks (Sitzmann et al., 2020) that adaptively generate parameters have further improved prediction performances in classification (Ratzlaff and Fuxin, 2019), super-resolution (Klocek et al., 2019) and image generation (Dupont et al., 2022; Koyuncu et al., 2023). We here adopt a hypernetwork-based INR implementation in UNIC to achieve more stable and faster model training for counting. Compared with existed image generation model, our main focus is to model density maps as a stochastic process, which is continuous and enables uncertainty quantification (UQ).

Uncertainty in counting: In recent years, researchers (Oh et al., 2020; Ranjan et al., 2020; Wang et al., 2022) have attempted to develop crowd-counting models with the uncertainty quantification capability. Oh et al. (2020) proposed a scalable neural network framework with quantification of decomposed uncertainty using a bootstrap ensemble. Ranjan et al. (2020) modeled the crowd density values using Gaussian distributions and developed a CNN architecture to predict these distributions, where an active sample selection strategy guided by the quantified uncertainty is used for reducing the amount of labeled data for training. To the best of our knowledge, UNIC is the first uncertainty-aware INR-based counting method based on density map estimation, where we predict the density distribution for any continuous spatial location that scale and generalize better for counting tasks across different image resolution.

3 UNIC

We propose UNIC, an object counting method utilizing hypernetwork-based implicit neural representation (INR), which models density maps as stochastic processes for accurate prediction and uncertainty quantification. We first briefly review the density map estimation (DME) based counting method in Section 3.1 and then introduce our continuous modeling of density map with INR and hypernetwork in Section 3.2. We further model the density maps as stochastic processes to quantify the uncertainties in object counting tasks 3.3, with the variational inference based training algorithm detailed in Section 3.4.

3.1 DME-based Counting

For a given image $\mathbf{I}$, let the counting annotation map $\mathcal{D}_{\mathbf{I}} = \{(\mathbf{z}_n, y_n)\}_1^N$ with a set of N labeled objects,

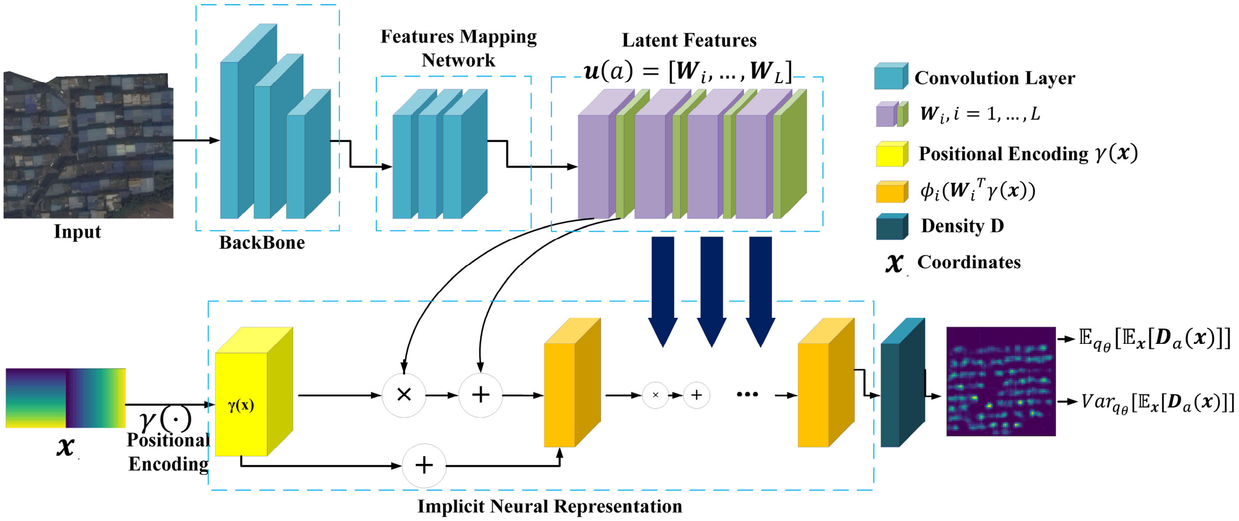
Input
BackBone
Features Mapping Network
Latent Features
$\boldsymbol{u}(a) = [\boldsymbol{W}_i, \ldots, \boldsymbol{W}_L]$
Convolution Layer
$\boldsymbol{W}_i, i = 1, \ldots, L$
Positional Encoding $\gamma(\boldsymbol{x})$
$\phi_i(\boldsymbol{W}_i{}^T\gamma(\boldsymbol{x}))$
Density D
$\boldsymbol{x}$ Coordinates
$\boldsymbol{x}$
$\gamma(\cdot)$
Positional Encoding
γ(x)
$\mathbb{E}_{q_\theta}[\mathbb{E}_{\boldsymbol{x}}[\boldsymbol{D}_a(\boldsymbol{x})]]$
$Var_{q_\theta}[\mathbb{E}_{\boldsymbol{x}}[\boldsymbol{D}_a(\boldsymbol{x})]]$
Implicit Neural Representation

Figure 1: Schematic diagram of UNIC. Given an input image $\mathbf{I}_t$, the predicted latent features of the encoder network are used to parameterize the decoder network for the test image, which predicts the count number $\mathbb{E}_{q_{\boldsymbol{\theta}}}[\mathbb{E}_{\mathbf{x}}[\mathbf{D}_t(\mathbf{x})]]$ as well as the associated uncertainty $\mathrm{Var}[\mathbb{E}_{\mathbf{x}}[\mathbf{D}_t(\mathbf{x})]]$ given the positional encoding of an arbitrary location $\boldsymbol{\gamma}(\mathbf{x}), \mathbf{x} \in [0,1]^2$.

where $\mathbf{z}_n$ denotes the normalized image-coordinate-based position of the n-th object, $\mathbf{z}_n \in [0,1]^2$ and $y_n = n$ is the corresponding label for each object. We focus on DME-based counting and this annotation map is used to derive the object density map $\mathbf{D}$ by convolution using a Gaussian kernel based on annotated object positions (Lempitsky and Zisserman, 2010; Wang et al., 2015; Fu et al., 2015; Paul Cohen et al., 2017; Gao et al., 2020):

$$\begin{aligned}\mathbf{D}^{gt}(\mathbf{x}_m) &\overset{\text{def}}{=} \sum_{n=1}^{N} \mathcal{N}\left(\mathbf{x}_m; \mathbf{z}_n, \sigma^2 \mathbf{1}_{2\times 2}\right) \\ &= \sum_{n=1}^{N} \frac{1}{\sqrt{2\pi}\sigma} \exp\left(-\frac{\|\mathbf{x}_m - \mathbf{z}_n\|_2^2}{2\sigma^2}\right),\end{aligned} \tag{1}$$

where $\mathbf{x}_m$ denotes the mth normalized pixel coordinate in the estimated density map with M pixels, $\mathbf{x}_m \in [0,1]^2$, $\mathcal{N}\left(\mathbf{x}_m; \mathbf{z}_n, \sigma^2 \mathbf{1}_{2\times 2}\right)$ denotes the 2D Gaussian distribution with the mean $\mathbf{z}_\mathrm{n}$ and isotropic covariance matrix $\sigma^2 \mathbf{1}_{2\times 2}$.

3.2 Density Maps as Continuous Functions

Instead of typical discrete density maps in existing DME-based counting methods (Lempitsky and Zisserman, 2010; Wang et al., 2015; Fu et al., 2015; Paul Cohen et al., 2017; Gao et al., 2020), UNIC leverages INR to predict a continuous density function. Such a model design can better scale up DME-based counting and reduce the counting bias due to potential blurring effects based on the input image resolution. To be more specific, we represent the estimated density map of image $\mathbf{I}_a$ as a continuous function $\mathcal{H}_a : [0,1]^2 \to \mathbb{R}_{\geq 0}$. The discrete pixel-based density map can be obtained by evaluating $\mathcal{H}_a$ over the discrete grids $\{\mathbf{x}_m\}_{m=1}^{M}$. This INR-based representation is not only more efficient and scalable than the pixel-based representation as detailed in the latter sections, but also allows us to generate the density map of arbitrarily high resolution by increasing the sampling frequency without the need to re-train the model.

In order to efficiently represent the unique density function by learning the corresponding density map prediction model $\mathcal{H}_a$ for each image $\mathbf{I}_a$, we first map the input image $\mathbf{I}_a$ into a latent space through an encoder network $f_{\boldsymbol{\theta}}(\cdot)$, and condition our density function $\mathcal{H}_a$ on latent representations $\mathbf{u}_a = f_{\boldsymbol{\theta}}(\mathbf{I}_a) \in \mathbb{R}^{d_c \times d_w \times d_h}$. Specifically, with $\mathcal{H}_a$ parameterized with a decoder network $h(\cdot)$, we utilize the latent representations $\mathbf{u}_a$ via a hypernetwork-based implementation:

$$\mathcal{H}_a(\cdot) = h(\cdot, \mathbf{u}_a) = h(\cdot, f_{\boldsymbol{\theta}}(\mathbf{I}_a)). \tag{2}$$

One main challenge for adopting INR for the continuous density representation is "spectral bias" (Tancik et al., 2020), a phenomenon that the neural network tends to learn only low-frequency features. To predict the density map that retains the details of the object location and size, we follow the previous works (Tancik et al., 2020) and use the Fourier encoding for the coordinate positions. Given a coordinate position $\mathbf{x} = [x^1, x^2]^T$, the positional encoding for $\mathbf{x}$:

$\boldsymbol{\gamma}(\mathbf{x})$ is given as follows:

$$\begin{aligned}\gamma^1(x^i) &= [\sin 2^0\pi x^i, \ldots, \sin 2^J\pi x^i]^T, i \in \{1,2\},\\ \gamma^2(x^i) &= [\cos 2^0\pi x^i, \ldots, \cos 2^J\pi x^i]^T, i \in \{1,2\},\\ \boldsymbol{\gamma}(\mathbf{x}) &= [\gamma^1(x^1)^T, \gamma^2(x^1)^T, \gamma^1(x^2)^T, \gamma^2(x^2)^T]^T,\end{aligned}$$

where log-linear spaced frequencies are used for each dimension, and the degree J is a hyperparameter determined by given tasks and datasets.

Note that we would like to derive a unique continuous density function via $\mathcal{H}_a$ for each image $\boldsymbol{I}_a$. In our hypernetwork-based model, instead of using a decoder network $h_{\boldsymbol{\psi}}(\cdot)$ parameterized by some learnable parameters $\boldsymbol{\psi}$ taking coordinate positions $\mathbf{x}$ and latent features $\mathbf{u}_a$ as two separate inputs, it is more efficient to directly use the latent features $\mathbf{u}_a$ to parameterize the decoder as a hypernetwork (Klocek et al., 2019). In this case, the latent features $\mathbf{u}_a$ can be rewritten as $\mathbf{u}_a = [\boldsymbol{W}_1, \ldots, \boldsymbol{W}_L]$, and the density at coordinate position $\mathbf{x}$ has the following expression, with the hypernetwork-based decoder being a Multi-Layer Perceptron (MLP) neural network:

$$\mathcal{H}_a(\mathbf{x}) = \boldsymbol{\phi}_L(\boldsymbol{W}_L^T \ldots \boldsymbol{\phi}_1(\boldsymbol{W}_1^T \boldsymbol{\gamma}(\mathbf{x}))), \qquad (3)$$

where L is the number of the decoder network layers, and $\boldsymbol{\phi}_l(\cdot)$ is the activation function at layer l. Here we use the same notation $\boldsymbol{W}_l$ to denote a 3-D matrix and a 2-D matrix by flattening its last two dimensions. We provide a schematic diagram of our UNIC framework in Figure 1.

We would like to emphasize that because of our hypernetwork-based continuous decoder implementation, it is easy to sample at arbitrary spatial locations through the learned density map prediction model $\mathcal{H}_a(\cdot)$ dependent on the given input image. This enables flexible stochastic training algorithms that can take the best advantage of available information in the training data and scale up DME-based counting by sampling arbitrary-size coordinates for gradient estimates to update model parameters.

3.3 Uncertainty-aware Counting

In many real-world applications, we want to develop a density map prediction model that provides count estimations and quantifies the associated uncertainties. This motivates us to model the density map $\mathbf{D}(\cdot)$ as two-dimensional random process, rather than estimating only a deterministic density function. To learn the posterior distribution of $\mathbf{D}(\cdot)$ in a Bayesian paradigm, we assume a variational distribution $q_{\boldsymbol{\theta}}(\mathbf{D}_a(\cdot)|\mathbf{I}_a)$ parameterized by $\boldsymbol{\theta}$ through $h(\cdot, f_{\boldsymbol{\theta}}(\mathbf{I}_a))$. With mild independence assumptions, minimizing the KL divergence between the variational distribution $q_{\boldsymbol{\theta}}(\mathbf{D}_a(\cdot)|\mathbf{I}_a)$ and ground-truth density maps $p(\mathbf{D}_a(\cdot)|\mathbf{I}_a, \mathcal{D}_{\mathbf{I}_a})$ ($\mathcal{D}_{\mathbf{I}_a}$ denotes the annotation map for image $\mathbf{I}_a$) can be approximately solved by maximizing the following derived Evidence Lower BOund (ELBO):

$$\begin{aligned}&\frac{1}{A}\sum_{a=1}^{A}\log p(\mathcal{D}_{\mathbf{I}_a}|\mathbf{I}_a)\\ \geq&\frac{1}{A}\sum_{a=1}^{A}[\log p(\mathbf{D}_a^{gt}(\cdot)|h(\cdot, f_{\boldsymbol{\theta}}(\mathbf{I}_a)), \mathbf{I}_a)\\ &- D_{KL}(q_{\boldsymbol{\theta}}(\mathbf{D}_a(\cdot)|\mathbf{I}_a)\|p_\eta(\mathbf{D}_a(\cdot)|\mathbf{I}_a))].\end{aligned} \qquad (4)$$

Here $p_\eta(\mathbf{D}_a(\cdot)|\mathbf{I}_a)$ is the prior distribution of $\mathbf{D}_a(\cdot)$, for which we can assume an uninformative prior, independent of $\mathbf{I}_a$ without loss of generality. We assume the likelihood $p(\mathbf{D}_a^{gt}(\cdot)|\boldsymbol{\theta}, \mathbf{I})$ implicitly defined by the discrepancy between the contribution of $\mathbf{D}_a^{gt}$ to the n-th object label $c_{n,a}^{gt}(\cdot)$ and the estimated contribution $c_{n,a}(\cdot)$, which has the following form:

$$p(\mathbf{D}_a^{gt}(\cdot)|\boldsymbol{\theta}, \mathbf{I}) \propto \prod_{i=1}^{N}\exp(-\|c_{i,a}(\cdot) - c_{i,a}^{\text{gt}}(\cdot)\|_2^2). \qquad (5)$$

We detail the definitions and derivations of $c_{n,a}^{gt}(\cdot)$ and $c_{n,a}(\cdot)$ in *Supplementary Materials* Section A.1.

3.4 Stochastic Estimation of Expected ELBO

By Jensen's inequality, the expression (4) can be further lower bounded by the following expected ELBO:

$$\begin{aligned}\mathbb{E}_{\mathbf{x}\sim p(\mathbf{x})}[\frac{1}{A}\sum_{a=1}^{A}&[\log p(\mathbf{D}_a^{gt}(\mathbf{x})|h(\cdot, f_{\boldsymbol{\theta}}(\mathbf{I}_a)), \mathbf{I}_a)\\ &- D_{KL}(q_{\boldsymbol{\theta}}(\mathbf{D}_a(\mathbf{x})|\mathbf{I}_a)\|p_\eta(\mathbf{D}_a(\mathbf{x})|\mathbf{I}_a))]],\end{aligned} \qquad (6)$$

where $p(\mathbf{x})$ is any probability distribution of $\mathbf{x}$. This optimization objective makes it possible to train our model with any existing stochastic optimization algorithm. Without loss of generality, we can choose $p(\mathbf{x})$ to be a uniform distribution $\mathbf{x} \sim \text{Uniform}[0,1]^2$. By further integrating (13) into (6) and deriving estimates by samples $\{\mathbf{x}_m\}_{m=1}^{K}$, we reach the following minimization objective:

$$\begin{aligned}\mathcal{L}_{ELBO} =&\frac{1}{A}\sum_{a=1}^{A}[\sum_{n=0}^{N}\mathbb{E}_{\mathbf{x}\sim\text{Unif}[0,1]^2}\left[\|c_{n,a}(\mathbf{x}) - c_{n,a}^{\text{gt}}(\mathbf{x})\|_2^2\right]\\ &+ \lambda\sum_{a=1}^{A}D_{KL}(q(\mathbf{D}_a(\mathbf{x})|\mathbf{I}_a)\|p_\eta(\mathbf{D}_a(\mathbf{x}))|\mathbf{I}_a))]\\ \approx&\frac{1}{K}\sum_{m=1}^{K}\sum_{a=1}^{A}[\sum_{n=0}^{N}\|c_{n,a}(\mathbf{x}_m) - c_{n,a}(\mathbf{x}_m)^{\text{gt}}\ \|_2^2\\ &+ \lambda D_{KL}(q(\mathbf{D}_a(\mathbf{x}_m)|\mathbf{I}_a)\|p_\eta(\mathbf{D}_a(\mathbf{x}_m))|\mathbf{I}_a))],\end{aligned} \qquad (7)$$

where λ is a hyperparameter reflecting our belief between the observed data and prior, following Higgins et al. (2016) and $n = 0$ denoting the background pixel modeling in Ma et al. (2019). While we sample $\mathbf{x}_m$ uniformly from an $S \times S$ image grid, sampling strategies with better efficiency can be developed with the prior knowledge of object locations. We will also show in Section 4.6 that the model trained with a larger S can make density map prediction with better accuracy and visual quality, which could be potentially useful for developing high precision detection models leveraging the derived density maps.

Given the optimized variational parameters $\boldsymbol{\theta}^*$ obtained through model training and a test image $\mathbf{I}_t$, we derive count predictions and estimate the uncertainties using the expectation $\mathbb{E}_{q_{\boldsymbol{\theta}}}\left[\mathbb{E}_{\mathbf{x}}\left[\mathbf{D}_t(\mathbf{x})\right]\right]$ and variance $\mathrm{Var}\left[\mathbb{E}_{\mathbf{x}}\left[\mathbf{D}_t(\mathbf{x})\right]\right]$ of the predicted count distribution respectively, which can be easily estimated with the samples from $q_{\boldsymbol{\theta}}$ over the normalized image domain.

4 EXPERIMENTS

We evaluate our UNIC on the Remote Sensing Object Counting (RSOC) dataset (Gao et al., 2020), whose data splits and summary statistics of "building" subset are provided in Section 4.1. We include the evaluation metrics for both counting accuracy and uncertainty quantification in Section 4.2, and training details with chosen hyper-parameters in Section 4.3. We compare our UNIC to baseline models and also visualize several examples of the predicted density maps derived by UNIC along with those by the baseline models in Section 4.4. We further evaluate the potential counting UQ capability in Section 4.5. Last but not least, we also provide ablation studies analyzing the sensitivity of different hyperparameters and sampling grids in our model in Sections 4.6.

4.1 Dataset

The RSOC dataset (Gao et al., 2020) is a large-scale image dataset for remote sensing count estimation, which contains $3,057$ 512×512 images with a total of $286,539$ objects labeled with "buildings", "ships" "small vehicles" and "large vehicles". We focus primarily on estimating the number of buildings in this paper and split $2,468$ images with an average of 30.3 building labels into training and test sets following Gao et al. (2020). The training set contains $1,205$ images and the test set contains $1,263$ images.

4.2 Evaluation Metrics

Evaluate Counting Accuracy: In our experiments, different counting methods are assessed using two commonly utilized metrics: the Mean Absolute Error (MAE) and the Root Mean Squared Error (RMSE) defined as follows:

$$\begin{aligned} MAE &= \frac{1}{A}\sum_{a=1}^{A}\left|C_{\mathbf{I}_a} - C_{\mathbf{I}_a}^{predict}\right|, \\ RMSE &= \sqrt{\frac{1}{A}\sum_{a=1}^{A}\left|C_{\mathbf{I}_a} - C_{\mathbf{I}_a}^{predict}\right|^2}, \end{aligned} \tag{8}$$

where $C_{\mathbf{I}_a}$ is the number of objects in $\mathbf{I}_a$, A is the total number of test images, and $C_{\mathbf{I}_a}^{predict}$ denotes the predicted number of objects in $\mathbf{I}_a$, calculated by the expectation $\mathbb{E}_{q_{\boldsymbol{\theta}}}\left[\mathbb{E}_{\mathbf{x}}\left[\mathbf{D}_a(\mathbf{x})\right]\right]$.

Evaluate Uncertainty Quantification: A well-calibrated model means that the predicted distribution by the model matches perfectly with the true data distribution. Here we use the *Calibration Curve* (Kuleshov et al., 2018) to measure the calibration performance of our predicted uncertainty in UNIC. Intuitively, in a regression setting, the frequency of the label to fall in the $\alpha\%$ predicted confidence interval should also be approximately $\alpha\%$. Formally, we say that the forecaster $C^{predict}$ is well calibrated if

$$q \overset{\text{def}}{=} \sum_{i=1}^{T} \frac{\mathbb{I}\left\{C_{\mathbf{I}_t}^{predict} \leq F_t^{-1}(p)\right\}}{T} = p, \forall p \in [0,1], \tag{9}$$

where T denotes the number of images on the test dataset; $\mathbb{I}(x) = 1$ when $x = True$, and $\mathbb{I}(x) = 0$ when $x = False$; $F_{\mathbf{I}_t}(\cdot)$ is the cumulative probability distribution (CDF) with $F_{\mathbf{I}_t}^{-1}(p)$ denoting the corresponding interval at the p confidence level. Denote the value of q as the observed confidential level based on our uncertainty-aware count estimation. Good uncertainty quantification will have q close to p.

4.3 Model Settings and Training Details

We compare our UNIC with ASPDNet (Gao et al., 2020) and PSGCNet (Gao et al., 2022), which are two state-of-the-art (SOTA) counting models for remote-sensing images, based on both MSE and Bayesian loss functions (Ma et al., 2019) for model training. For fair comparison, we use the same VGG-16 (Simonyan and Zisserman, 2014) backbone for all of the ASPDNet, PSGCNet and UNIC models. Our feature mapping network consists of three convolution layers to generate latent features and max-pooling layers to reduce the

[1] We here adopt the sampling based Bayesian counting loss as detailed in the text instead of the original loss in Ma et al. (2019).

Table 1: Means and standard deviation values of counting accuracy on different random model running with different input image sizes on the RSOC dataset. Our UNIC model consistently outperforms the SOTA counting models for remote sensing images with different input image resolution. UNIC is consistently the best performing model with their prediction errors highlighted in the **bold** font.

Method	Loss		64× 64		256× 256		512× 512	
	MSE.	Bayes.	MAE	RMSE	MAE	RMSE	MAE	RMSE
PSGCNet	✓		8.90±0.36	12.69±0.47	7.33±0.34	11.02±0.22	7.41±0.24	11.16±0.27
PSGCNet		✓	7.92±0.27	11.43±0.21	7.18±0.11	10.98±0.13	7.30±0.16	10.91±0.22
ASPDNet	✓		9.33±0.40	12.54±0.34	7.40±0.24	11.06±0.31	7.27±0.14	10.68±0.22
ASPDNet		✓	8.19±0.19	12.07±0.11	7.59±0.19	11.07±0.25	7.21±0.13	10.58±0.27
UNIC		✓[1]	**7.63±0.11**	**11.30± 0.08**	**6.83±0.23**	**10.42± 0.24**	**7.11±0.08**	**10.53±0.11**

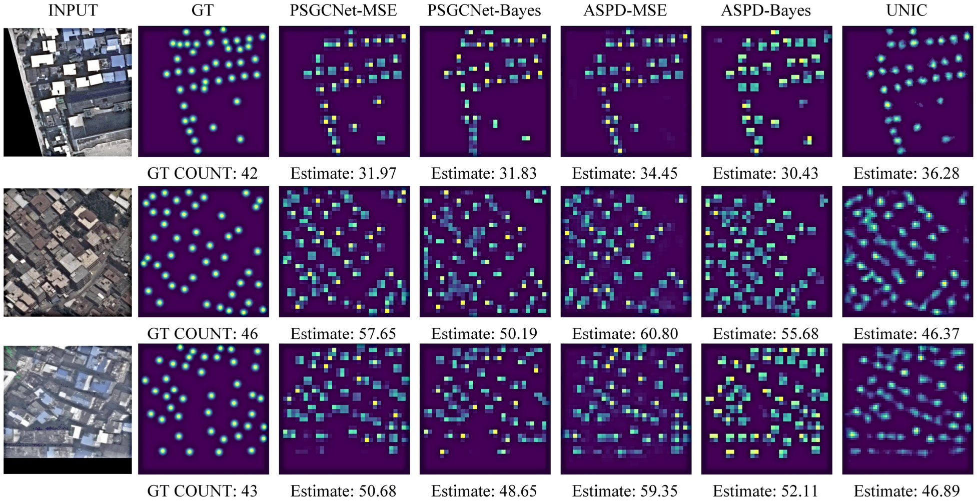

Figure 2: Predicted density maps by UNIC and other baseline models of three test images from RSOC. Three test images are randomly sampled (INPUT) with the ground truth (GT). Density maps are generated by PSGCNet with MSE loss (PSGCNet+MSE), PSGCNet with Bayesian counting loss (PSGCNet+Bayes), ASPDNet with MSE loss (ASPD+MSE), ASPDNet with Bayesian counting loss (ASPD+Bayes), and UNIC (UNIC). Warmer colors denote higher values while cooler colors denote lower values.

Table 2: Model complexity by the number of parameters with different input image sizes on the RSOC dataset.

Method	# of Param.		
	64 × 64	256 × 256	512 × 512
PSGCNet	$\approx 28.1M$	$\approx 28.1M$	$\approx 28.1M$
ASPDNet	$\approx 27.5M$	$\approx 27.5M$	$\approx 27.5M$
UNIC	$\approx 13.0M$	$\approx 18.0M$	$\approx 18.1M$

size of feature maps. Our decoder network in UNIC consists of four fully connected layers with residual connection whose weights and biases are from latent features, and one fully connected layer with learnable parameters to output mean and standard deviation of variational posterior, with the detailed parameterization provided in *Supplementary Materials* Section B.

We use the Adam (Kingma and Ba, 2014) optimizer for all the competing models with the learning rate of $1e-4$. For simplicity, we adopt Gaussian distributions for both prior and variational distributions. We initialize our model parameters using random samples from a Gaussian distribution $\mathcal{N}(0, 0.01^2)$, and set $\sigma = \frac{1}{64}$ and $J = 100$ unless specified. Especially, we set $\sigma = \frac{1}{16}$ when input size = 64 to generate reasonable density maps, we set $S = 64$ in Sections 4.4 and 4.5, and $S = 32$ in Section 4.6 to visualize the effect of J and σ. Except for the results in Table 1, we set input image size to 256 × 256 for all the other experiments, while the downsampling ratio is set to 8 in all baseline methods. We augment the training data by randomly flipping the input images both horizontally and vertically. All the experimental results are obtained on a workstation with a NVIDIA V100 32GB GPU. The best performing models that have the lowest MSE with proper density maps in the first 1000 training epochs are selected for reporting the results. We run our experiments multiple times using the random seeds 0, 1, 64, and 123 and report the average performance for reproducibility.

4.4 Comparison with Baseline Models

We compare the counting accuracy by UNIC and the baseline models with different input image resolutions on the RSOC dataset while keeping the model backbone fixed.

Table 1 reports the average counting accuracy along with the standard deviation values over four random runs, from which it can be observed that UNIC consistently achieves the best counting performance on the RSOC building benchmark with different resolutions. UNIC exhibits significant enhancements over the second-best performing ASPDNet with the Bayesian Loss and reduces MAE by 4.87% and RMSE by 5.10% for the input resolution 256×256. While the counting performance for all the tested models typically degrades with downsampled input images, due to the potential information loss of image details, UNIC not only maintains the overall best performance but also provides more stable counting prediction accuracy with different input resolutions compared to other baseline models. Compared to the case with input size 256×256, we observe larger prediction errors when the input size is set to be 512×512. This is possibly due to *spectral bias*(Tancik et al., 2020; Sitzmann et al., 2020), a phenomenon that the neural network tends to learn only low-frequency features. Although more information is provided with the increasing input resolution, the model focuses more on low-frequency signals during training, which brings limited benefit to the final counting accuracy.

We also provide several examples from the test images to visually compare the derived density maps by UNIC and other baseline methods, including PSGCNet and ASPDNet, trained with both the traditional MSE and Bayesian counting loss functions. From Figure 2, the baseline methods do not perform as well as our UNIC when the objects' appearance or illumination is complex. For example, in the first image, ASPDNet misses some buildings in the bottom half of the image. UNIC also has fewer false positives in highly dense object regions. In the second and third images, regions (Zhang et al., 2022) where boundaries between buildings and background are hard to distinguish appear in both ASPDNet and PSGCNet results, causing typically larger count estimates than the ground-truth labels. Compared with other methods, thanks to the ability of INR to capture more detailed image information, UNIC outputs higher-quality density maps while achieving lower counting errors. To make the over-estimation and under-estimation regions clearer, we also provide information gain(IG) (Kümmerer et al., 2015) maps in *Supplementary Materials* Section C.

Moreover, our hypernetwork-based decoder contains drastically less parameters compared to the baseline models, which is shown in Table 2. For example, our UNIC with the VGG-16 backbone only uses $\sim 18.0M$ model parameters when input size = 256×256, which is 34.5% less than ASPDNet models, with $\sim 27.5M$ parameters. For the case that the input size is 64×64, we set $d_w = d_h = 8$ instead of 16, which makes the model much smaller but still has constant performance. Besides model complexity, for deriving the density map given a 256×256 input image on our workstation, our UNIC only takes 3.6ms, 75.8% less compared to ASPDNet which takes 14.9ms, and 20.0% less compared to PSGCNet which takes 4.5ms. Our UNIC can achieve better computational and parameter efficiency and scales better with larger input images.

4.5 Uncertainty Quantification

Our UNIC model also provides reasonable uncertainty estimation in terms of both informativeness to possible erroneous prediction and prediction calibration. To see this, we provide the scatter plot of the MAE prediction error with respect to the quantified uncertainty of each test image by UNIC in Figure 3, as well as the calibration plot in Figure 4.

We can see in Figure 3 that the prediction error is mostly small when UNIC derives confident counting prediction. As the predicted uncertainty increases, the model also exhibits a higher chance to make a count prediction deviated from the ground-truth label. This trend indicates the informativeness of the quantified uncertainty to potential prediction error, which could be important to safety-critical applications. We can also observe in Figure 4 that the calibration curve of our model is reasonably close to the $y = x$ line, indicating that the predicted $\alpha\%$ confidence interval covers approximately $\alpha\%$ of the ground truth of the test data. The deviation of the calibration curve from the perfect $y = x$ line at the end is possibly caused by our Gaussian variational distribution, which allows negative density map prediction and does not match perfectly with reality. Strategies like *post-hoc* calibration or choosing a more complicated variational distribution can be applied to address this mismatch and potentially increase the uncertainty quantification quality.

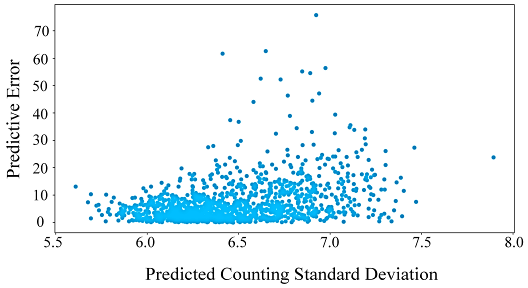

Figure 3: Scatter of predictive error with respect to the predicted counting standard deviation. The model exhibit a higher chance to make a erroneous prediction as the predicted standard deviation increase.

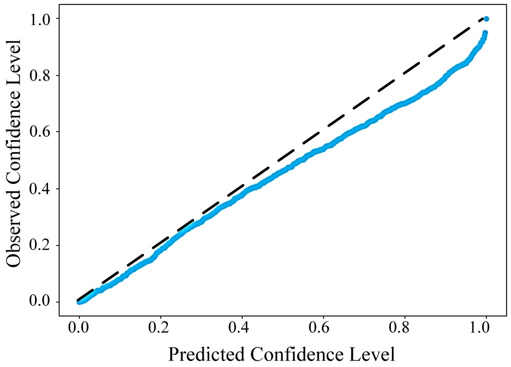

Figure 4: Calibration curve of the predicted uncertainty. The $x-$axis represents the predicted confidence interval and the $y-$axis represents the frequency of the ground-truth counts falling into the predicted interval. A well-calibrated model should have the calibration curve close to $y = x$.

4.6 Ablation Studies

We further evaluate the sensitivity of the counting performance of our UNIC model with respect to three hyperparameters: 1) J, which is the hyperparameter controlling the degree or dimension of the positional encoding; 2) σ, which is the hyperparameter for generating the ground-truth density map; and 3) S, the size of predefined grids from which we sample our density function for training.

Effect of J: We report the counting accuracy with different J on the RSOC dataset in Table 3. The best-performing model is achieved when J is set to be 100. In Figure 5 we show three exemplar density maps with different J along with the ground-truth density maps. We observe sharper density maps and better counting performance as J increases from 16 to 100.

Effect of σ: In Figure 6 we show the density maps predicted by UNIC using two test images from RSOC with $\sigma = \frac{1}{64}$ and $\sigma = \frac{1}{32}$. We can see smaller σ values lead to sharper density maps compared with larger σ. UNIC also achieves consistent counting performance with respect to different σ values.

Table 3: Counting accuracy with different J for positional encoding on RSOC.

J (Input Size = 256)	MAE	RMSE
16	7.19	10.67
32	7.10	10.56
64	6.92	10.39
80	6.95	10.43
100	6.85	10.31

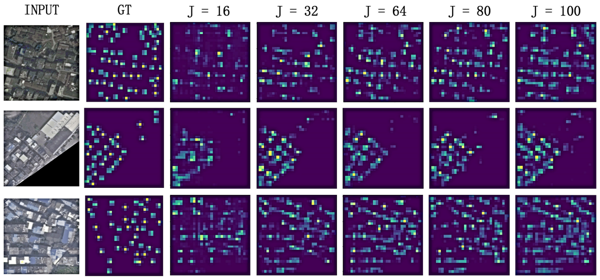

Figure 5: Predicted density maps with different J along with the ground-truth density maps of three test images from RSOC. Given example test images (INPUT), it is observed that with increasing J from left to right, UNIC can capture more detailed image information with large enough J and derive sharper density maps, similar as the ground truth (GT).

Effect of S**:** We also report the prediction accuracy and include the predicted density maps by our UNIC trained with the loss estimated by sampling from the grids of different size S in Table 4 and Figure 7. As S increases from 32×32 to 128×128, our model can predict better density maps in terms of visual quality. A possible explanation is that given a large enough number of samples T, we are providing the model with the ground-truth density closer to a continuous 2D function as S increases. This will allow the model to capture the subtle boundary information helpful for deriving the density map with better visual quality. We note that the counting prediction accuracy is stable with different S, indicating that with a sufficient number of density function samples, UNIC can be trained to achieve consistent superior counting performance.

5 CONCLUSION

In this paper, we develop UNIC with a hypernetwork-based INR implementation for object counting, which achieves the state-of-the-art counting performance and

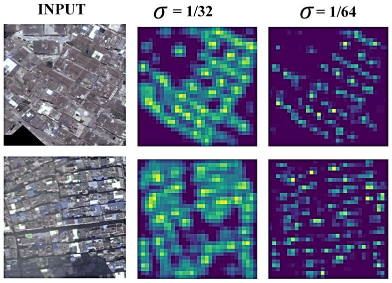

Figure 6: Predicted density maps with different σ values for two test images from RSOC.

Table 4: Counting accuracy with different sampling grid sizes for training S on the RSOC dataset.

S (resolution of density map)	MAE	RMSE
32 × 32	6.85	10.31
64 × 64	6.83	10.42
128 × 128	6.82	10.32

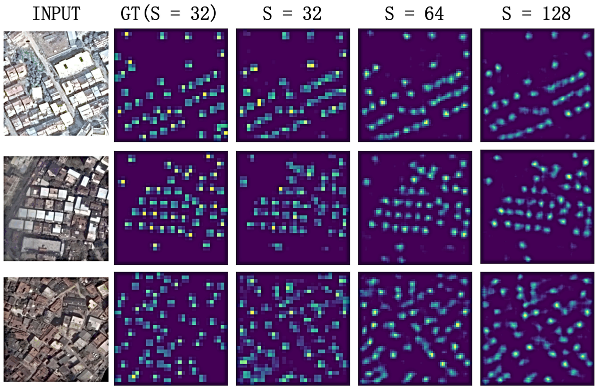

Figure 7: Predicted density maps by UNIC trained with the loss estimated by sampling from grids of different size S. Given example test images (INPUT), it is observed that with increasing S from left to right, UNIC can derive closer density maps to the ground truth (GT) while maintaining consistent counting accuracy.

enables uncertainty quantification. With efficient model training by designing a sampling based Bayesian counting loss function for stable and faster convergence, UNIC has demonstrated superior counting performance compared to existing methods but with significantly reduced number of model parameters on the RSOC dataset. Our experiments have showcased the benefits of our proposed INR-based continuous decoder as well as the sampling based Bayesian loss in UNIC, which has the potential to be easily incorporated into existing counting methods and other image analysis tasks for further performance improvement.

Acknowledgements

This work was supported in part by the Department of Energy (DOE) Award DE-SC0012704, and the National Science Foundation (NSF) Award IIS-2212419.

References

Barrowclough, O. J., Muntingh, G., Nainamalai, V., and Stangeby, I. (2021). Binary segmentation of medical images using implicit spline representations and deep learning. *Computer Aided Geometric Design*, 85:101972.

Chan, A. B., Liang, Z.-S. J., and Vasconcelos, N. (2008). Privacy preserving crowd monitoring: Counting people without people models or tracking. In *2008 IEEE Conference on Computer Vision and Pattern Recognition*, pages 1–7. IEEE.

Dupont, E., Whye Teh, Y., and Doucet, A. (2022). Generative models as distributions of functions. In Camps-Valls, G., Ruiz, F. J. R., and Valera, I., editors, *Proceedings of The 25th International Conference on Artificial Intelligence and Statistics*, volume 151 of *Proceedings of Machine Learning Research*, pages 2989–3015. PMLR.

Fu, M., Xu, P., Li, X., Liu, Q., Ye, M., and Zhu, C. (2015). Fast crowd density estimation with convolutional neural networks. *Engineering Applications of Artificial Intelligence*, 43:81–88.

Gao, G., Liu, Q., Hu, Z., Li, L., Wen, Q., and Wang, Y. (2022). PSGCNet: A pyramidal scale and global context guided network for dense object counting in remote-sensing images. *IEEE Transactions on Geoscience and Remote Sensing*, 60:1–12.

Gao, G., Liu, Q., and Wang, Y. (2020). Counting from sky: A large-scale data set for remote sensing object counting and a benchmark method. *IEEE Transactions on Geoscience and Remote Sensing*, 59(5):3642–3655.

Higgins, I., Matthey, L., Pal, A., Burgess, C., Glorot, X., Botvinick, M., Mohamed, S., and Lerchner, A. (2016). beta-VAE: Learning basic visual concepts with a constrained variational framework. In *International Conference on Learning Representations*.

Kingma, D. P. and Ba, J. (2014). Adam: A method for stochastic optimization. *arXiv preprint arXiv:1412.6980*.

Klocek, S., Maziarka, Ł., Wołczyk, M., Tabor, J., Nowak, J., and Śmieja, M. (2019). Hypernetwork functional image representation. In *International Conference on Artificial Neural Networks*, pages 496–510. Springer.

Koyuncu, B., Sanchez-Martin, P., Peis, I., Olmos, P. M., and Valera, I. (2023). Variational mixture of hypergenerators for learning distributions over functions. *arXiv preprint arXiv:2302.06223*.

Krizhevsky, A., Sutskever, I., and Hinton, G. E. (2012). Imagenet classification with deep convolutional neural networks. In Pereira, F., Burges, C., Bottou, L., and Weinberger, K., editors, *Advances in Neural Information Processing Systems*, volume 25. Curran Associates, Inc.

Kuleshov, V., Fenner, N., and Ermon, S. (2018). Accurate uncertainties for deep learning using calibrated regression. In *International Conference on Machine Learning*, pages 2796–2804. PMLR.

Kümmerer, M., Wallis, T. S., and Bethge, M. (2015). Information-theoretic model comparison unifies saliency metrics. *Proceedings of the National Academy of Sciences*, 112(52):16054–16059.

Lempitsky, V. and Zisserman, A. (2010). Learning to count objects in images. *Advances in Neural Information Processing Systems*, 23.

Li, Y., Zhang, X., and Chen, D. (2018). CSRNet: Dilated convolutional neural networks for understanding the highly congested scenes. In *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition*, pages 1091–1100.

Lin, S.-F., Chen, J.-Y., and Chao, H.-X. (2001). Estimation of number of people in crowded scenes using perspective transformation. *IEEE Transactions on Systems, Man, and Cybernetics-Part A: Systems and Humans*, 31(6):645–654.

Ma, Z., Wei, X., Hong, X., and Gong, Y. (2019). Bayesian loss for crowd count estimation with point supervision. In *Proceedings of the IEEE/CVF International Conference on Computer Vision*, pages 6142–6151.

Ma, Z., Yu, L., and Chan, A. B. (2015). Small instance detection by integer programming on object density maps. In *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition*, pages 3689–3697.

Mescheder, L., Oechsle, M., Niemeyer, M., Nowozin, S., and Geiger, A. (2019). Occupancy networks: Learning 3d reconstruction in function space. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, pages 4460–4470.

Oechsle, M., Mescheder, L., Niemeyer, M., Strauss, T., and Geiger, A. (2019). Texture fields: Learning texture representations in function space. In *Proceedings of the IEEE/CVF International Conference on Computer Vision*, pages 4531–4540.

Oh, M.-h., Olsen, P., and Ramamurthy, K. N. (2020). Crowd counting with decomposed uncertainty. In *Proceedings of the AAAI Conference on Artificial Intelligence*, volume 34, pages 11799–11806.

Paul Cohen, J., Boucher, G., Glastonbury, C. A., Lo, H. Z., and Bengio, Y. (2017). Count-ception: Counting by fully convolutional redundant counting. In *Proceedings of the IEEE International Conference on Computer Vision workshops*, pages 18–26.

Ranjan, V., Wang, B., Shah, M., and Hoai, M. (2020). Uncertainty estimation and sample selection for crowd counting. In *Proceedings of the Asian Conference on Computer Vision*.

Ratzlaff, N. and Fuxin, L. (2019). HyperGAN: A generative model for diverse, performant neural networks. In *International Conference on Machine Learning*, pages 5361–5369. PMLR.

Simonyan, K. and Zisserman, A. (2014). Very deep convolutional networks for large-scale image recognition. *arXiv preprint arXiv:1409.1556*.

Sitzmann, V., Martel, J., Bergman, A., Lindell, D., and Wetzstein, G. (2020). Implicit neural representations with periodic activation functions. *Advances in Neural Information Processing Systems*, 33:7462–7473.

Skorokhodov, I., Ignatyev, S., and Elhoseiny, M. (2021). Adversarial generation of continuous images. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, pages 10753–10764.

Stanley, K. O. (2007). Compositional pattern producing networks: A novel abstraction of development. *Genetic Programming and Evolvable Machines*, 8:131–162.

Tancik, M., Srinivasan, P., Mildenhall, B., Fridovich-Keil, S., Raghavan, N., Singhal, U., Ramamoorthi, R., Barron, J., and Ng, R. (2020). Fourier features let networks learn high frequency functions in low dimensional domains. *Advances in Neural Information Processing Systems*, 33:7537–7547.

Wan, J. and Chan, A. (2019). Adaptive density map generation for crowd counting. In *Proceedings of the IEEE/CVF International Conference on Computer Vision*, pages 1130–1139.

Wang, C., Zhang, H., Yang, L., Liu, S., and Cao, X. (2015). Deep people counting in extremely dense crowds. In *Proceedings of the 23rd ACM international conference on Multimedia*, pages 1299–1302.

Wang, Y., Gu, M., Zhou, M., and Qian, X. (2022). Attention-based deep bayesian counting for ai-augmented agriculture. In *Proceedings of the 20th ACM Conference on Embedded Networked Sensor Systems*, pages 1109–1115.

Zhang, L., Li, J., Zhang, S., and Wang, Z. (2022). An improved Bayesian loss function for crowd counting. In *2022 International Conference on Cyber-Physical Social Intelligence (ICCSI)*, pages 631–636. IEEE.

Checklist

1. For all models and algorithms presented, check if you include:

   (a) A clear description of the mathematical setting, assumptions, algorithm, and/or model. [Yes]

   (b) An analysis of the properties and complexity (time, space, sample size) of any algorithm. [No]

   (c) (Optional) Anonymized source code, with specification of all dependencies, including external libraries. [Yes]

2. For any theoretical claim, check if you include:

   (a) Statements of the full set of assumptions of all theoretical results. [Not Applicable]

   (b) Complete proofs of all theoretical results. [Not Applicable]

   (c) Clear explanations of any assumptions. [Not Applicable]

3. For all figures and tables that present empirical results, check if you include:

   (a) The code, data, and instructions needed to reproduce the main experimental results (either in the supplemental material or as a URL). [Yes]

   (b) All the training details (e.g., data splits, hyperparameters, how they were chosen). [Yes]

   (c) A clear definition of the specific measure or statistics and error bars (e.g., with respect to the random seed after running experiments multiple times). [Yes]

   (d) A description of the computing infrastructure used. (e.g., type of GPUs, internal cluster, or cloud provider). [Yes]

4. If you are using existing assets (e.g., code, data, models) or curating/releasing new assets, check if you include:

   (a) Citations of the creator If your work uses existing assets. [Yes]

   (b) The license information of the assets, if applicable. [Not Applicable]

   (c) New assets either in the supplemental material or as a URL, if applicable. [Not Applicable]

   (d) Information about consent from data providers/curators. [Yes]

   (e) Discussion of sensible content if applicable, e.g., personally identifiable information or offensive content. [Not Applicable]

5. If you used crowdsourcing or conducted research with human subjects, check if you include:

   (a) The full text of instructions given to participants and screenshots. [Not Applicable]

   (b) Descriptions of potential participant risks, with links to Institutional Review Board (IRB) approvals if applicable. [Not Applicable]

   (c) The estimated hourly wage paid to participants and the total amount spent on participant compensation. [Not Applicable]

A SUPPLEMENTARY INFORMATION FOR MODEL TRAINING

A.1 Derivation of Log-likelihood $\log p(\mathbf{D}_a^{gt}(\cdot)|h(\cdot, f_{\boldsymbol{\theta}}(\mathbf{I}_a)), \mathbf{I}_a)$

Here we describe how we derive $\log p(\mathbf{D}_a^{gt}(\cdot)|h(\cdot, f_{\boldsymbol{\theta}}(\mathbf{I}_a)), \mathbf{I}_a)$ with Bayesian-loss-like likelihood in the Section 3.3 of the *Main Text*. We omit the subscript a for image index for simplicity. We make similar assumptions as Ma et al. (2019) and assume a random function $y(\cdot) : \mathbb{R}^2 \rightarrow \{1, \cdots, N\}$, with $y(\mathbf{x})$ denoting whether the location $\mathbf{x}$ belongs to one of the N objects with a prior distribution $p(y(\cdot))$. The posterior probability given annotations $\mathcal{D}_{\mathbf{I}} = \{(\mathbf{z}_n, y_n)\}_1^N$ at an arbitrary location $\mathbf{x}$ can be derived by Bayes' rule:

$$p\left(y(\cdot) \mid \mathcal{D}_{\mathbf{I}}\right) = \frac{p\left(\mathcal{D}_{\mathbf{I}} \mid y(\cdot)\right) p\left(y(\cdot)\right)}{p\left(\mathcal{D}_{\mathbf{I}}\right)} = \frac{p\left(\mathcal{D}_{\mathbf{I}} \mid y(\cdot)\right) p\left(y(\cdot)\right)}{\sum_{i=1}^{N} p\left(\mathcal{D}_{\mathbf{I}} \mid y(\cdot) = i\right) p\left(y(\cdot) = i\right)}. \tag{10}$$

Here we adopt a discrete uniform prior $p\left(y(\cdot)\right) = \frac{1}{N}$ for all $y(\cdot)$ and assume the Gaussian likelihood with the isotropic covariance matrix $p\left(\mathcal{D}_{\mathbf{I}} \mid y(\mathbf{x}) = n\right) = \mathcal{N}\left(\mathbf{x}; \mathbf{z}_n, \sigma^2 \mathbf{1}_{2\times 2}\right)$. We further simplify the posterior distribution:

$$p\left(y(\cdot) = n \mid \mathcal{D}_{\mathbf{I}}\right) = \frac{\mathcal{N}\left(\cdot; \mathbf{z}_n, \sigma^2 \mathbf{1}_{2\times 2}\right)}{\sum_{i=1}^{N} \mathcal{N}\left(\cdot; \mathbf{z}_i, \sigma^2 \mathbf{1}_{2\times 2}\right)}. \tag{11}$$

The contribution of $\mathbf{D}^{gt}$ to n-th label at each location can be computed as follows:

$$\begin{aligned} c_n^{gt}(\cdot) &= p\left(y(\cdot) = n \mid \mathcal{D}_{\mathbf{I}}\right) \mathbf{D}^{\text{gt}}\ (\cdot) \\ &= \frac{\mathcal{N}\left(\cdot; \mathbf{z}_n, \sigma^2 \mathbf{1}_{2\times 2}\right)}{\sum_{i=1}^{N} \mathcal{N}\left(\cdot; \mathbf{z}_n, \sigma^2 \mathbf{1}_{2\times 2}\right)} \times \sum_{i=1}^{N} \mathcal{N}\left(\cdot; \mathbf{z}_n, \sigma^2 \mathbf{1}_{2\times 2}\right) \\ &= \mathcal{N}\left(\cdot; \mathbf{z}_n, \sigma^2 \mathbf{1}_{2\times 2}\right), \end{aligned} \tag{12}$$

which is a Gaussian density function centered at $\mathbf{z}_n$. We assume the likelihood $p(\mathbf{D}^{gt}(\cdot)|\boldsymbol{\theta}, \mathbf{I})$ implicitly defined by the discrepancy between $c_n^{gt}(\cdot)$ and estimated contribution $c_n(\cdot) = p\left(y(\cdot) = n \mid \mathcal{D}_{\mathbf{I}}\right) \mathcal{H}_a(\cdot)$, which has the following form:

$$p(\mathbf{D}^{gt}(\cdot)|\boldsymbol{\theta}, \mathbf{I}) \propto \prod_{i=1}^{N} \exp(-\|c_i(\cdot) - c_i^{\text{gt}}(\cdot)\|_2^2). \tag{13}$$

B SUPPLEMENTARY INFORMATION FOR NETWORK PARAMETERIZATION

Given the latent features $\mathbf{u}_a \in \mathbb{R}^{d_c \times d_w \times d_h}$, we first split $\mathbf{u}_a$ into $\boldsymbol{F}_1, \ldots, \boldsymbol{F}_L$, with each $\boldsymbol{F}_l \in \mathbb{R}^{d_c(l) \times d_w \times d_h}$, where $d_c(l)$ is the number of channels of $\boldsymbol{F}_l$ with $\sum_{l=1}^{L} d_c(l) = d_c$. Each $\boldsymbol{F}_l$ is further split into the weight matrix $\boldsymbol{W}_l \in \mathbb{R}^{d_c(l) \times d_w d_h}$ and bias vector $b_l \in \mathbb{R}^{1 \times d_w d_h}$ with the last two dimensions flattened into one. We predict the density at coordinate position $\mathbf{x}$ with the following expression:

$$\mathcal{H}_a(\mathbf{x}) = H_{W_L, b_L}(\ldots H_{W_1, b_1}(\boldsymbol{\gamma}(\mathbf{x}))), \tag{14}$$

where each $H_{W_l, b_l}(\cdot)$ is a fully connected layer with activation function and residual connection. We always set $d_c(1) = 4J$ and $d_c(l) = d_w d_h, l = 2, \ldots, L$ to match the dimensionality. We set $d_w = d_h = 8$ when the input size is 64×64 and $d_w = d_h = 16$ for input sizes 256×256 and 512×512.

C INFORMATION GAIN MAPS

Given a coordinate position $\mathbf{x}_m = [x_m^1, x_m^2]^T$ on images $\mathbf{I}_a$, predictions of the model $\mathbf{D}_a$, and the ground truth $\mathbf{D}_a^{gt}$. $\mathbf{D}_a$ and $\mathbf{D}_a^{gt}$ are firstly divided by the baseline model (prior) to get the "image-based prediction" map (Kümmerer et al., 2015). Both maps are then log-transformed and multiplied by the ground truth to calculate information gains $\mathbf{IG}_a^{model}$ and $\mathbf{IG}_a^{gt}$. Subtracting the standard information gain $\mathbf{IG}_a^{gt}$ from the model's information gain $\mathbf{IG}_a^{model}$ yields a difference map $\mathbf{IG}_a^{dif}$ of the possible information gain. The calculation of $\mathbf{IG}_a^{dif}$ has the following form:

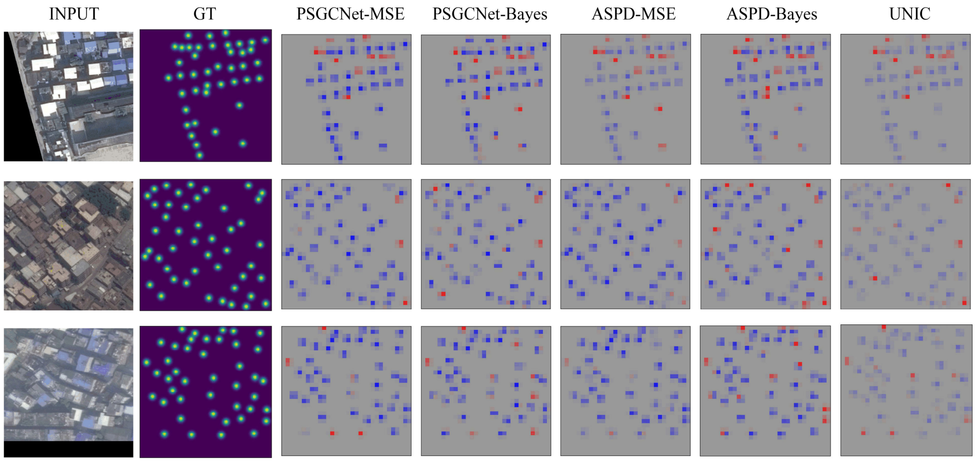

Figure 8: Information Gain (IG) maps by UNIC and baseline models of three test images from RSOC. Three test images are the same as the ones in Figure 2. IG maps for baselines are generated by PSGCNet with MSE loss (PSGCNet+MSE), PSGCNet with Bayesian counting loss (PSGCNet+Bayes), ASPDNet with MSE loss (ASPD+MSE), ASPDNet with Bayesian counting loss (ASPD+Bayes), and UNIC (UNIC), where blue regions denote over-estimation regions and red regions denote under-estimation regions. We can see UNIC makes fewer errors than other methods.

Table 5: Results on the CARPK dataset.

Method	MAE	RMSE
ASPD-MSE	5.77	8.56
ASPD-Bayes	8.63	10.31
PCSG-MSE	5.48	7.98
PCSG-Bayes	6.94	9.30
UNIC	**5.37**	**7.65**

$$\mathbf{IG}_a^{dif} = \log(\mathbf{D}_a/prior)\mathbf{D}_a^{gt} - \log(\mathbf{D}_a^{gt}/prior)\mathbf{D}_a^{gt}. \tag{15}$$

In our experiments, we set a 2D Gaussian distribution map as the prior, whose mean equals 0 and standard deviation equals 32.

$\mathbf{IG}_a^{dif}$ clearly shows where and by how much the model's predictions fail. In this case, the positive part in $\mathbf{IG}_a^{dif}$ shows the over-estimation region, and the negative part in $\mathbf{IG}_a^{dif}$ shows the under-estimation region.

We provide $\mathbf{IG}_a^{dif}$ in Figure 8, where the test images are the same as the ones in Figure 2. We can see that UNIC has less over- or under-estimation compared with the baseline methods.

D EXPERIMENTS ON CARPK

We also test our methods on the Car Parking Lot Dataset (CARPK), which has 1448 images with 90,000 cars from 4 different parking lots. The image resolution is 1280 × 720.

In this experiment, we resize all images into 512 × 512, four baseline methods have the same setting as the experiments on the RSOC dataset. Since UNIC has better counting performance on processing 256 × 256 input images, we randomly crop 256 × 256 pieces from the training images as our training data. For the test dataset, We separate the test images into four 256 × 256 pieces and predict their counting number independently, then add them together to get the total counting number.

The counting performance is shown in table 5, we can see Unic has the best counting performance compared with four baseline models.