

Accurate Gigapixel Crowd Counting by Iterative Zooming and Refinement

Arian Bakhtiarnia, Qi Zhang, and Alexandros Iosifidis

Abstract—The increasing prevalence of gigapixel resolutions has presented new challenges for crowd counting. Such resolutions are far beyond the memory and computation limits of current GPUs, and available deep neural network architectures and training procedures are not designed for such massive inputs. Although several methods have been proposed to address these challenges, they are either limited to downsampling the input image to a small size, or borrowing from other gigapixel tasks, which are not tailored for crowd counting. In this paper, we propose a novel method called GigaZoom, which iteratively zooms into the densest areas of the image and refines coarser density maps with finer details. Through experiments, we show that GigaZoom obtains the state-of-the-art for gigapixel crowd counting and improves the accuracy of the next best method by 42%.

I. INTRODUCTION

Crowd counting has many applications in video surveillance, social safety and crowd analysis and is an active area of research in the literature [1]. Since most crowd counting applications and datasets use surveillance footage, the input to crowd counting models are high-resolution images, typically Full HD (1,920×1,080 pixels) or even higher. Gigapixel resolutions can capture and process much more detail than previously possible, and are recently becoming more widespread [2]. However, working with gigapixel resolutions presents several unique challenges. Modern high-end GPUs are not capable of fitting gigapixel images in memory or processing such high resolutions in reasonable time. Furthermore, the architectures of deep neural networks are not designed to receive such massive images as input.

Recently, several methods have been proposed for crowd counting on gigapixel images. However, these methods either use the simplest solution, which is to downsample the input gigapixel to a manageable resolution before processing, or borrow from gigapixel literature in other deep learning tasks. The issue with the latter approach is that gigapixel methods for other deep learning tasks such as object detection or cancer detection do not tackle unique challenges present in crowd counting, such as reliance on global information and sensitivity to perspective. On the other hand, the proposed method called *GigaZoom* is tailored to crowd counting and is thus able to obtain significantly more accurate results compared to previous

Arian Bakhtiarnia, Qi Zhang and Alexandros Iosifidis are with DIGIT, the Department of Electrical and Computer Engineering, Aarhus University, Aarhus, Midtjylland, Denmark (e-mail: arianbakh@ece.au.dk; qz@ece.au.dk; ai@ece.au.dk).

This work was funded by the European Union’s Horizon 2020 research and innovation programme under grant agreement No 957337, and by the Danish Council for Independent Research under Grant No. 9131-00119B.

methods. GigaZoom works by iteratively zooming into the densest areas of the image and refining the coarser density map with finer details. Our code is publicly available¹.

The rest of this paper is organized as follows. Section II summarizes the related work in crowd counting and gigapixel deep learning literature. Section III presents the proposed method. Section IV describes the experimental setup and provides experimental results as well as ablation studies. Finally, section V concludes the paper by summarizing contribution and results, and providing directions for future research.

II. RELATED WORK

A. Crowd Counting

The goal of crowd counting is to count the total number of people present in a given input image [1]. The input to crowd counting models is an image or a video frame, and the output is a density map showing the crowd density at each location of the image. The values in the density map can be summed up to obtain a single number representing the total number of people in the image. Widely used crowd counting datasets contain high resolution images, for instance, images in Shanghai Tech Part A and Part B datasets [3] have average resolutions of 868×589 and 1,024×768 pixels, respectively, and images in the UCF-QNRF dataset [4] have an average resolution of 2902×2013 pixels. However, these resolutions are much lower than gigapixel resolutions. At the time of this writing, PANDA [5] is the only publicly available dataset for gigapixel crowd counting. PANDA contains 45 images with resolutions up to 26,908×15,024 pixels taken from three different scenes: an airport terminal, a graduation ceremony, and a marathon. Images in the PANDA dataset are extremely densely populated with crowd sizes of up to 4,302 people, and ground truth annotations are available in the form of bounding boxes for each person’s head. PANDA offers no predefined training or test splits.

Various crowd counting methods exist in the literature. CSRNet [6] uses the first ten layers of VGG-16 [7], pre-trained on ImageNet [8], as a feature extractor, which is followed by six dilated convolution layers to produce the output density map. Gigapixel CSRNet [9] utilizes CSRNet to process gigapixel images. During the training phase, CSRNet is trained on image patches of size 1,920×1,200 pixels, taken across three scales: the original gigapixel image, as well as the gigapixel image downsampled to $\frac{1}{16}$ and $\frac{1}{64}$ of the original size. In the inference phase, image patches of the same size are

¹<https://gitlab.au.dk/maleci/gigazoom>

passed on to the trained CSRNet in non-overlapping sliding windows to produce a density map for each scale. The three density maps are then averaged to obtain a single aggregated density map.

PromptMix [10] downsamples gigapixel images to $2,560 \times 1,440$ pixels, then processes them using CSRNet. It improves the accuracy of CSRNet by mixing artificially generated data with real data during training. SASNet [11] is a high-performing crowd counting method on various popular datasets such as Shanghai Tech and UCF-QNRF. Similar to CSRNet, SASNet also uses the first ten layers of VGG-16 [7], pre-trained on ImageNet [8], as feature extractor, and fuses features extracted by these layers across multiple scales to obtain an accurate density map.

B. Gigapixel Deep Learning

The term ‘‘gigapixel’’ suggests an image containing one billion pixels. However, images with resolutions ranging from 100 megapixels up to hundreds of gigapixels are considered to be ‘‘gigapixel images’’ in the literature [2]. Using gigapixel images and videos reveals much more detail about the scene and has the potential to significantly improve the accuracy of deep learning tasks. However, as previously mentioned, processing gigapixel images with deep learning is challenging due to GPU memory and computation limits. Even without considering GPU limits, existing deep learning architectures and methods are not capable of properly training the massive number of parameters that would result from using gigapixel images directly as input. Moreover, gigapixel datasets typically contain a very low number of images, since manually labelling such large images is a difficult task. For instance, the PANDA dataset contains only 45 examples compared to the 1,535 examples UCF-QNRF.

The most common approach for dealing with very high resolutions in deep learning is to downsample the images to a manageable resolution. However, this obscures details and negates the benefits of capturing gigapixel images. For instance, as shown in Figure 1, there are locations in the downsampled gigapixel image where several people are represented by a single pixel, making it impossible for a deep learning model to accurately predict crowd density.

Processing gigapixel whole-slide images (WSIs) is common in histopathology for cancer detection, detecting metastasis (the spread of cancer), neuropathology and detecting tissue components [2]. For instance, HIPT [12] processes gigapixel WSIs using a hierarchy of Vision Transformers, and [13] uses neural image compression on WSIs so they can be processed with a CNN on a single GPU. However, a key difference between histopathology and crowd counting is the lack of perspective in the former. This means that in WSIs, cells and tissues always have roughly the same size, whereas in gigapixel crowd counting, the bounding box for a person near the camera can be up to 1 million times larger than that of a person far away.

Several methods exist for gigapixel object detection. For instance, GigaDet [14] is a near real-time object detection

method for gigapixel videos. GigaDet counts the number of objects on regions of downsampled version of image across multiple scales, then processes the top candidate regions to detect objects. However, as explained in section III, gigapixel object detection methods cannot be directly used for crowd counting.

III. GIGAZOOM

GigaZoom is inspired by how people act when they are asked to count the number of people in gigapixel images, where they zoom into the dense regions of the crowd until they can distinguish individuals. Similarly, GigaZoom iteratively zooms into multiple dense regions to refine the coarse density map. Section III-A provides the details of the zooming and refinement process, and section III-B describes how the multiple regions are detected.

A. Iterative Zooming and Replacing

Iterative zooming and replacing consists of two steps: a forward pass that iteratively zooms into the densest area of the image, and a backward pass that combines the density maps obtained during the forward pass to construct the final density map. Figure 2 shows an overview of the forward pass. Given a gigapixel image I_0 of resolution $w_0 \times h_0$, we perform L zoom-in operations until we reach a resolution within GPU memory limits. Note that L is a hyper-parameter of the method. The location of the zoomed-in image I_{t+1} depends on the density map obtained by previous image I_t . Since resolution of I_t is beyond the GPU memory limit for $t < L$, we are not able to use I_t directly as input to the crowd counting model. Therefore, we first need to downsample I_t to $w_{\max} \times h_{\max}$, defined as the maximum image resolution that can fit into the available GPU memory.

The width and height of I_t are determined based on the zoom formula. *Linear zoom* is defined as

$$h_t = h_0 - \left(\frac{h_0 - h_{\max}}{L} \right) t, \quad w_t = w_0 - \left(\frac{w_0 - w_{\max}}{L} \right) t; \quad (1)$$

whereas *exponential zoom* is defined as

$$h_t = h_0 \left(\frac{h_{\max}}{h_0} \right)^{\frac{t}{L}}, \quad w_t = w_0 \left(\frac{w_{\max}}{w_0} \right)^{\frac{t}{L}}. \quad (2)$$

Suppose that we have performed t zoom-ins so far and obtained image I_t within I_0 , where (O_t^w, O_t^h) is the top left corner of I_t inside I_0 . Since the width and height of I_{t+1} are already known based on the zoom formula, our goal is to determine (O_{t+1}^w, O_{t+1}^h) , which is the top left corner of I_{t+1} inside I_0 . We start by uniformly downsampling I_t to I_t^{small} with a resolution of $w_{\max} \times h_{\max}$. We then pass I_t^{small} to a crowd counting model to obtain density map D_t . Note that the density map size $w_{\max}^D \times h_{\max}^D$ might be smaller in size than I_t^{small} due to pooling operations in the crowd counting model.

The density of all sub-images within I_t , which are candidates for I_{t+1} , can be calculated using a simple convolution

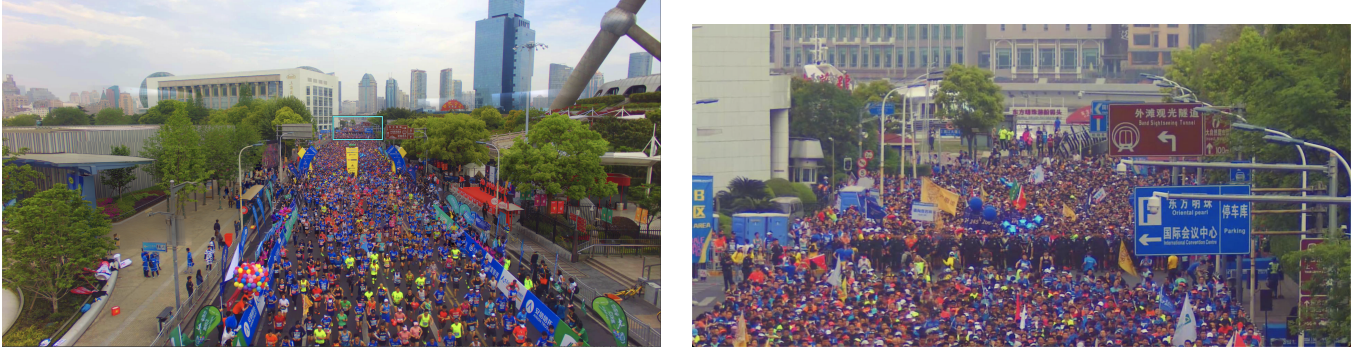


Fig. 1: (left) Example gigapixel image from the PANDA dataset, with a resolution of $26,908 \times 15,024$ downsampled to $2,688 \times 1,412$; and (right) zoomed into the region specified by the rectangle in the original image, with a resolution of $2,880 \times 1,410$ pixels.

operation on the obtained density map, with an all-ones kernel of size $k_w \times k_h$, where

$$k_w = \left(\frac{w_{t+1}}{w_t} \right) w_{\max}^D, \quad k_h = \left(\frac{h_{t+1}}{h_t} \right) h_{\max}^D. \quad (3)$$

In the resulting matrix S_t , the point $(O_{t,D}^w, O_{t,D}^h)$ with the maximum value corresponds to the sub-image with the highest density. The top left corner of I_{t+1} can then be determined based on

$$\frac{O_{t+1}^w - O_t^w}{w_t} = \frac{O_{t,D}^w}{w_{\max}^D}, \quad \frac{O_{t+1}^h - O_t^h}{h_t} = \frac{O_{t,D}^h}{h_{\max}^D}. \quad (4)$$

Figure 3 shows an overview of the backward pass. During the forward pass, the density maps D_t , $t = 0, \dots, L$ along with the region of D_t that corresponds to D_{t+1} are saved to be used in the backward pass. The backward pass starts by resizing the finest density map D_L and replacing the region of D_{L-1} that corresponds to D_L to obtain an improved density map D'_{L-1} . Subsequently, D'_{L-1} is resized and placed in the corresponding region in D_{L-2} , and this process is repeated until the final improved density map is obtained. We tested more complex merging operations than simply replacing regions of density maps, for instance, we trained a CNN to combine D_t and resized D'_{t+1} to obtain an estimation closer to the corresponding part of the ground truth density map. However, replacement always resulted in the highest accuracy. Another simple merging operation is averaging, which is used in Gigapixel CSRNet. However, taking the average of density maps across multiple scales is not sensible, since the more zoomed-in density maps are almost always more accurate.

Crowd counting models are designed and trained for a specific range of crowd density, therefore, if the density goes above or falls below that range, their error increases. Another advantage of GigaZoom over Gigapixel CSRNet is that by zooming into dense areas, it ensures that low density areas are not processed separately. In contrast, Gigapixel CSRNet always detects a small crowd of people even if the image patch is completely empty. This is exacerbated by the fact that in gigapixel images, many locations of the image are empty,

resulting in a massive error. Note that empty regions are not an issue for gigapixel object detection methods, since they would simply be ignored. However, since, the density maps are added together in crowd counting, the errors accumulate.

B. Multiple Zoom Regions

Iterative zooming and replacing only zooms into a single region. However, multiple dense regions might be present in a given image. Therefore, we specify several regions to apply iterative zooming and replacing. We start by smoothing the coarsest density map D_0 using a Gaussian filter to remove small spikes in density. Peaks in the smoothed density map are then detected using a local maximum filter [15]. The detected peaks are then filtered based on a threshold λ , and the remaining peaks are clustered using the k-means algorithm to k clusters. Finally, we apply iterative zooming and replacing on sub-images centered at the cluster centers. The overall process is depicted in Figure 4.

Note that using multiple zoom regions may lead to conflicts since some areas might be processed during several iterative zooming and replacing operations. To resolve these conflicts, we tested several aggregation strategies such as averaging or using the maximum value. However, we found that all strategies obtain similar results. Therefore, we opted for the simplest strategy, which is to use the latest result in case of a conflict.

GigaZoom performs $k \times L$ CSRNet inferences per input gigapixel image. With the hyper-parameters specified in section IV, this translates to 20 CSRNet inferences, which is more than $10 \times$ faster than Gigapixel CSRNet, which performs an average of 204 CSRNet inferences per input gigapixel image. Note that iterative zooming and replacing cannot be parallelized, however, multiple iterative zooming and replacing operations can be performed in parallel.

IV. EXPERIMENTS

A. Setup and Results

Since PANDA [5] does not specify training and test splits, we selected 30 images for training, 6 images for validation

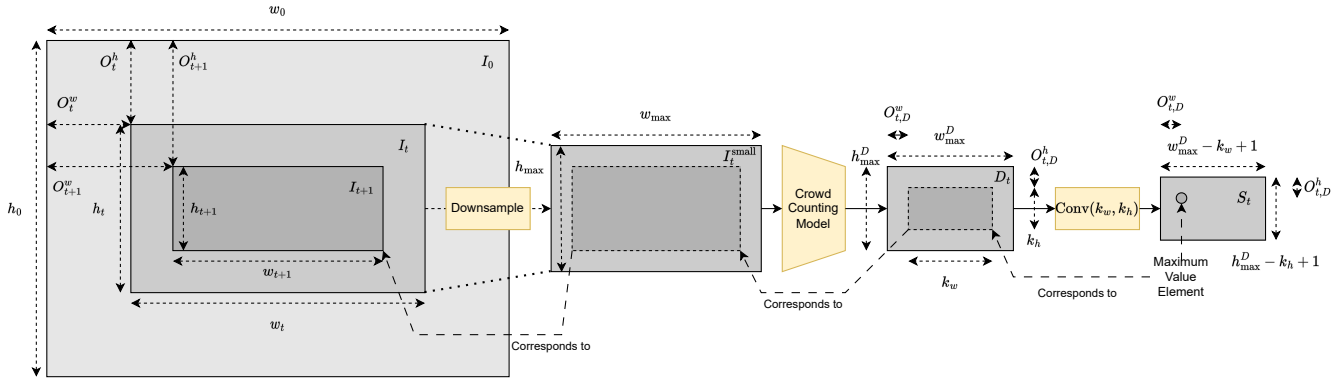


Fig. 2: Overview of the forward pass in iterative zooming and replacing.

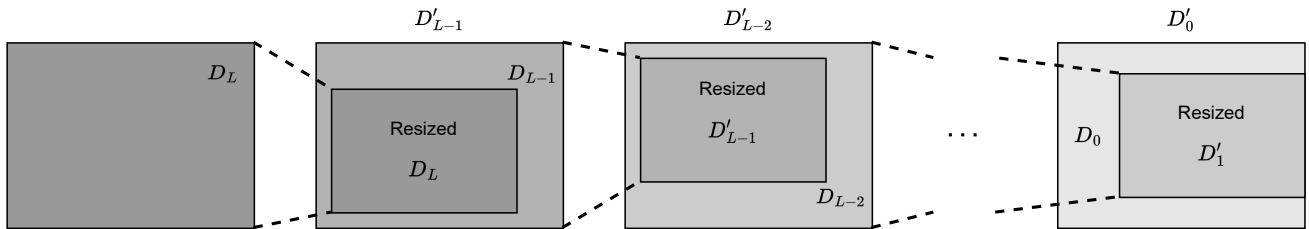


Fig. 3: Overview of the backward pass in iterative zooming and replacing.

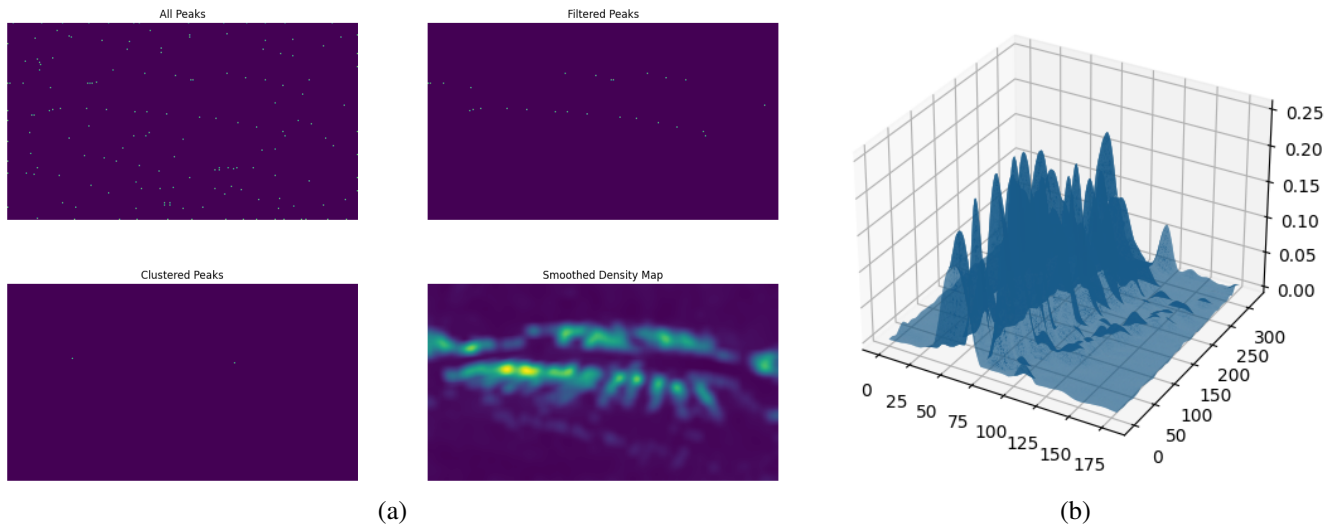


Fig. 4: (a) The smoothed density map as well as detected, filtered and clustered peaks; and (b) three-dimensional visualization of the smoothed density map.

Method	Year	MAE↓
Gigapixel CSRNet [9]	2019	2680.20
SASNet [11]	2021	263.88
PromptMix [10]	2023	110.34
GigaZoom (ours)	2023	63.51

TABLE I: Comparison of crowd counting performance for various methods on the PANDA gigapixel dataset.

and 9 images for test. The selection procedure can be viewed in our code. To obtain a ground truth density map from the bounding box annotations available in the PANDA dataset, for each bounding box, we apply a 2D Gaussian filter with $\sigma = 4$ and filter size the same as the bounding box.

The hyper-parameters used in GigaDet are as follows. We use exponential zoom with $L = 10$. The maximum resolution $w_{\max} \times h_{\max}$ that fits our GPU memory is $2,560 \times 1,440$. To determine multiple zoom regions, a Gaussian filter with $\sigma = 4$ and radius of 7 is used for smoothing, threshold $\lambda = 0.1$ is used for filtering and number of clusters $k = 2$ is used for clustering. We use two separate crowd counting models: a PromptMix model [10] to obtain D_0 , and for all other density maps D_1, \dots, D_L we use a CSRNet model [9] trained on patches of different scales. The first model is trained with the procedure outlined in [10], and the second model is trained by initializing with pre-trained weights from the PromptMix model, and fine-tuning for 100 epochs with a weight decay of 10^{-4} , batch size of 12 and a learning rate of 10^{-4} which is multiplied by 0.99 each epoch. All experiments were conducted on $3 \times$ Nvidia A6000 GPUs, each with 48 GBs of video memory.

Crowd counting methods are typically evaluated by using the mean absolute error (MAE) or the mean squared error (MSE), defined as

$$\text{MAE} = \frac{\sum_{i=1}^N |\hat{y}_i - y_i|}{N}, \quad \text{MSE} = \frac{\sum_{i=1}^N (\hat{y}_i - y_i)^2}{N}, \quad (5)$$

where \hat{y}_i is the prediction for i -th image, y_i is the ground truth label, and N is the total number of examples in the dataset. In crowd counting, MAE is typically used as a measure of accuracy, whereas MSE is a measure of robustness [16]. Since our primary objective is accuracy, we use MAE to evaluate crowd counting methods in this work.

The original SASNet paper does not include experiments on the PANDA dataset [11], therefore, we initialize the training with pre-trained weights for Shanghai Tech Part A and fine-tune on the PANDA dataset downsampled to $2,560 \times 1,440$ pixels. Although Gigapixel CSRNet uses the PANDA dataset, the authors do not report accuracy metrics, therefore, we reproduce the method to measure its accuracy. Since PromptMix includes experiments on PANDA, we use the number from the original paper. Table I compares the accuracy of GigaZoom with previous methods on the PANDA dataset. Observe that GigaZoom significantly outperforms other methods.

Zoom Method	MAE↓
Linear	81.02
Exponential	63.51

TABLE II: Effect of linear and exponential zoom on the accuracy of GigaZoom.

Zoom Levels (L)	MAE↓
5	80.04
10	64.49
20	105.45

TABLE III: Effect of the number of zoom levels on the accuracy of GigaZoom. Clustering was not used in these experiments, and only a single pass of iterative zooming and replacing was performed on the densest sub-image of the input.

B. Ablation Studies

Table II compares the accuracy obtained by the two different zoom methods defined in Equations 1 and 2, which shows that exponential zoom leads to a higher accuracy. Table III shows the effect of the number of zoom levels L on the accuracy. Based on these results, using too few or too many zoom levels can lead to sub-optimal accuracy. Table IV shows that using multiple zoom regions can slightly boost the accuracy. However, similar to the number of zoom levels, using too few or too many clusters can degrade the accuracy. Even though the accuracy improvement is slight in these experiments, using multiple zoom regions makes GigaZoom more robust and might lead to more significant improvements in other scenarios and scenes. We also investigated the effect of *overzooming* in Table V. Overzooming is defined as zooming beyond a 1-to-1 pixel ratio, where several pixels in the resulting image correspond to a single pixel in the original image, effectively upsampling a region of the original gigapixel image. However, these results show that the method does not benefit from overzooming.

Multiple Zoom Regions	Clusters	MAE↓
×	-	64.49
✓	1	70.19
✓	2	63.51
✓	5	79.34

TABLE IV: Effect of multiple zoom regions on the accuracy of GigaZoom.

Zoom Levels (L)	Overzoom Levels	MAE↓
10	0	64.49
10	1	67.97
10	2	93.26

TABLE V: Effect of overzooming on the accuracy of GigaZoom. Clustering was not used in these experiments, and only a single pass of iterative zooming and replacing was performed on the densest sub-image of the input.

V. CONCLUSION

We showed that our proposed method significantly outperforms existing methods for crowd counting on gigapixel images. Through ablation studies, we showed that exponential zoom performs better than linear, a moderate number of zoom levels achieves best accuracy, and using multiple zoom regions provides robustness for inputs with multiple dense crowds. Although GigaZoom is much more efficient than Gigapixel CSRNet, it still performs multiple CSRNet inferences, which can result in a long inference time overall.

Currently, PANDA is the only publicly available dataset for this task, which contains only 45 images taken from three scenes. In order to further compare and validate methods, it is crucial that more gigapixel crowd counting datasets are created and published, that have a higher number of examples taken from more diverse scenes.

REFERENCES

- [1] G. Gao, J. Gao *et al.*, “Cnn-based density estimation and crowd counting: A survey,” *arXiv*, 2020.
- [2] A. Bakhtiarnia, Q. Zhang, and A. Iosifidis, “Efficient high-resolution deep learning: A survey,” *arXiv*, 2022.
- [3] Y. Zhang, D. Zhou *et al.*, “Single-image crowd counting via multi-column convolutional neural network,” *CVPR*, 2016.
- [4] H. Idrees, M. Tayyab *et al.*, “Composition loss for counting, density map estimation and localization in dense crowds,” *ECCV*, 2018.
- [5] X. Wang, X. Zhang *et al.*, “Panda: A gigapixel-level human-centric video dataset,” *CVPR*, 2020.
- [6] Y. Li, X. Zhang, and D. Chen, “Csrnet: Dilated convolutional neural networks for understanding the highly congested scenes,” *CVPR*, 2018.
- [7] K. Simonyan and A. Zisserman, “Very deep convolutional networks for large-scale image recognition,” *ICLR*, 2015.
- [8] J. Deng, W. Dong *et al.*, “Imagenet: A large-scale hierarchical image database,” *CVPR*, 2009.
- [9] Z. Cao, R. Yan *et al.*, “Gigapixel-level image crowd counting using csrnet,” *ICMEW*, 2019.
- [10] A. Bakhtiarnia, Q. Zhang, and A. Iosifidis, “Promptmix: Text-to-image diffusion models enhance the performance of lightweight networks,” *IJCNN*, 2023.
- [11] Q. Song, C. Wang *et al.*, “To choose or to fuse? scale selection for crowd counting,” *AAAI*, 2021.
- [12] R. J. Chen, C. Chen *et al.*, “Scaling vision transformers to gigapixel images via hierarchical self-supervised learning,” *CVPR*, 2022.
- [13] D. Tellez, G. Litjens *et al.*, “Neural image compression for gigapixel histopathology image analysis,” *IEEE Transactions on Pattern Analysis and Machine Intelligence*, 2021.
- [14] K. Chen, Z. Wang *et al.*, “Towards real-time object detection in gigapixel-level video,” *Neurocomputing*, 2022.
- [15] M. Wolfer, K. Niemann, and D. G. Goodenough, “Local maximum filtering for the extraction of tree locations and basal area from high spatial resolution imagery,” *Remote Sensing of Environment*, 2000.
- [16] F. Dai, H. Liu *et al.*, “Dense scale network for crowd counting,” *ICMR*, 2021.