Dense Crowd Counting using Swin-UNet

Keshav Kumar Prabhakharan
University at Buffalo, SUNY
keshavku@buffalo.edu

Sumana Madhireddy
University at Buffalo, SUNY
sumanama@buffalo.edu

Abstract

In many real-world applications, such as event management and security surveillance, crowd counting is essential. In this project, we introduce a novel hybrid deep learning architecture that combines U-Net with Swin Transformers [1] for dense crowd counting. Our model effectively estimates crowd density and head counts in extremely congested areas by utilizing the ability of U-Net to capture fine-grained detail and the multi-scale hierarchical feature extraction capabilities of Swin Transformers. To guarantee robustness in various circumstances, we train and assess our model using a mix of popular datasets, such as ShanghaiTech [13], UCF-QNRF [4], and UCF-CC-50 [3]. Our application is made to assist public transportation authorities, law enforcement agencies, and event planners in efficiently managing and monitoring dense crowd areas.

1. Project Overview

This project aims to develop an advanced application that can reliably estimate the number of people in densely populated areas, addressing crucial needs in safety and resource management. For instance, consider the tragic incident at the Astroworld Festival in 2021, where a crowd surge resulted in multiple fatalities and injuries due to inadequate crowd control. By offering accurate density estimates, the application would enable better planning and quicker responses to prevent overcrowding and ensure the well-being of attendees. This tool is essential not only for managing large-scale public events but also for improving safety in various high-density scenarios, from concerts to transportation hubs.

The current state-of-the-art in crowd counting includes several advanced systems that excel in point recognition and feature extraction to accurately locate and count individuals in crowded environments. Some notable examples include FGENet [8], which employs a Fine-Grained Extraction Network coupled with a Three Task Combination loss function; PSL-Net [9], a Pseudo Square Label Network built on a VGG-based architecture; and VMambaCC [7], a Visual State Space Model that utilizes Feature Pyramids to enhance detection accuracy. On the other hand, density map-based approaches generate density maps from ground truth annotations and train models to predict these maps. The crowd count is obtained by summing all the values in the density maps. The state-of-the-art approaches with density map approaches are M-SFANet [10], which uses atrous spatial pyramid pooling (ASPP) and context-aware network (CAN); SANet [2], which uses scale aggregation networks; CSRNet [5], which uses a CNN for 2D feature extraction and another CNN with the dilated kernel to capture multi-scale features; MCNN [13], which uses multiple CNN branches with multiple kernel size to capture features at multiple-scales. While these methods have achieved promising results, they still face challenges in various scenarios, particularly with extremely dense crowds.

Our approach follows the density map-based approaches. It processes a densely crowded image through the Swin-UNet [1] architecture, which generates a predicted density map as an intermediate output. This density map is then used to estimate the total crowd count, effectively quantifying the number of people in crowded spaces.

In summary, by combining the advantages of Swin Transformers and U-Net topologies, this project aims to improve crowd counting in densely crowded places. With broad applications in public safety, event management, and transportation networks, our model provides a scalable and dependable way to monitor and control dense crowds.

Both authors contributed equally to the project, encompassing all aspects including research on approaches, dataset preparation, coding, reporting, and presentation.

2. Approach

The architecture of our Swin-UNet [1] model with the reinforcement network [12] is illustrated in Figure 1. In this architecture, the U-Net architecture performs well at maintaining fine-grained details and obtaining accurate localization, while the Swin Transformers [6] are effective at capturing multi-scale hierarchical characteristics. Our architecture can handle the complexity of dense crowd counting more accurately and robustly by combining these two mod-

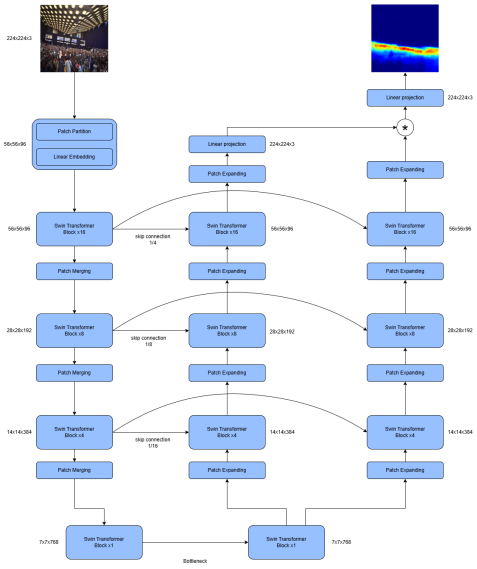

Figure 1. Swin-UNet with reinforcement network

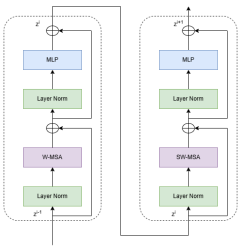

Figure 2. Two Successive Swin Transformer Blocks

els. The network has 3 branches, first the encoder branch, followed by a split in the network which flows parallel to the Density Map Estimation (DME) branch and the Reinforcement branch. The Encoder branch extracts the multi-scale feature maps and the DME branch outputs the density map. The reinforcement branch is used to construct an auxiliary input which helps the network converge faster and keep local pattern consistency. The dense crowd images are first divided into non-overlapping patches of size 4x4, generating embedding vectors with an embedding dimension of 96. These embeddings are then processed through a U-Net style encoder-decoder network with skip connections.

The Encoder consists of four Swin Transformer [6] layers, each followed by a patch merging layer. The first swin transformer layer contains 16 successive swin transformer blocks, the second swin transformer layer contains 8 successive swin transformer blocks, the third layer consists of 4 successive swin transformer blocks and the last layer (bottleneck layer) consists of 2 successive swin transformer blocks. As illustrated in Figure 2, the swin transformer block includes layer normalization followed by a windowed multi-headed self-attention (W-MSA/SW-MSA) module. A residual connection is added, followed by another layer normalization and a two-layer multi-layer perceptron (MLP). The Swin Transformer with patch merging is designed to learn hierarchical multi-scale features.

The decoder branches i.e., the DME and reinforcement branches also consist of four swin transformer layers followed by patch expansion. Like the encoder, each layer in the decoders includes 16, 8, and 4 Swin Transformer blocks to match the respective encoder layers. During patch expansion, skip connections from the corresponding encoder layers are concatenated, adhering to the U-Net architecture principles. This design allows the decoder to effectively reconstruct high-resolution details, such as the precise locations of head coordinates within the crowd images. By leveraging the hierarchical multi-scale features learned in the encoder, the decoder refines these details to produce accurate density maps.

The difference between the DME branch and the Reinforcement branch is that the Reinforcement branch feeds the final output from the decoder block to a conv 1×1×1 followed by Sigmoid activation. This generates the Reinforcement map which is element-wise multiplied by the final output from the decoder block of the DME branch. This is fed to the last conv 1×1×1 layer which generates the final density map.

The high-level encoder-decoder architecture with the reinforcement network was implemented independently while the individual blocks for swin-transformer, patch merging, and patch expansion were referred from the official Swin-UNet [1] implementation. The loss function, the dataset preparation, training, and evaluation blocks are all implemented independently.

3. Experimental Protocol

3.1. Datasets

We trained our model on the UCF-QNRF [4] dataset and evaluated our model on 3 different publicly available crowd-counting datasets including UCF-QNRF [4], ShanghatTech Part-A [13], UCF-CC-50 [3]. The evaluation results of our model on these datasets are presented in Section 4 below.

3.1.1 UCF-QNRF

The UCF-QNRF [4] dataset is currently the largest dataset available for training and evaluating crowd-counting and localization methods, based on the number of annotations. It consists of 1,535 images, split into 1,201 images for training and 334 images for testing. Our model was trained on the UCF-QNRF dataset, using 20% of the 1,201 training images for cross-validation.

3.1.2 ShanghaiTech Part-A

The ShanghaiTech [13] dataset is a large-scale crowd-counting dataset. It consists of 1198 annotated crowd images. The dataset is divided into two parts, Part-A containing 482 images and Part-B containing 716 images. Part A is split into train and test subsets consisting of 300 and 182 images, respectively. Part B is split into train and test subsets consisting of 400 and 316 images. Images from Part A were collected from the Internet and covered dense crowds, while images from Part B were collected on the busy streets of Shanghai with sparse crowds. We evaluated our model on the test subset of the ShanghaiTech Part-A dataset.

3.1.3 UCF-CC-50

UCF-CC-50 [3] is a dataset for crowd counting and consists of images of extremely dense crowds. It has 50 images with 63,974 head center annotations in total. The head counts range between 94 and 4,543 per image. The small dataset size and large variance make this a very challenging counting dataset. Hence we also evaluated our model on the UCF-CC-50 dataset.

3.2. Loss Function

For the loss function, we use a combined loss function of Mean Absolute Error (MAE) on the density maps and Binary Cross Entropy (BCE) loss on the reinforcement maps.

3.2.1 MAE Loss

The MAE loss is particularly suitable for density maps because it directly penalizes the absolute differences between predicted and ground truth densities, ensuring that both overestimates and underestimates contribute linearly to the loss. This linearity makes MAE more robust to outliers compared to losses like Mean Squared Error (MSE), which can disproportionately penalize large errors. The MAE loss function is given by the Equation 1.

$$\mathcal{L}_{\text{MAE}} = \frac{1}{N}\sum_{i=1}^{N}|\hat{y}_i - y_i| \tag{1}$$

where $\hat{y}_i$ is the predicted density value at pixel i, y_i is the target density value at pixel i and N is the total number of pixels.

3.2.2 BCE Loss

The reinforcement maps are binary classification maps where each pixel is labeled as either containing a head or not. The BCE loss quantifies the difference between the predicted probability and the true label for each pixel. This loss function drives the model to produce probabilities near 1 for pixels that contain a head and near 0 for those that do not, which is crucial for accurate localization. The BCE loss function is given by the Equation 2.

$$\mathcal{L}_{\text{BCE}} = \frac{1}{N}\sum_{i=1}^{N}[y_i \log(\hat{y}_i) + (1 - y_i)\log(1 - \hat{y}_i)] \tag{2}$$

where $\hat{y}_i$ is the predicted probability of pixel i , y_i is the target class at pixel i and N is the total number of pixels.

3.2.3 Combined Loss

The combined loss function leverages the strengths of both MAE and BCE losses. By applying MAE to the density maps, the model is encouraged to accurately estimate crowd densities, ensuring that the total count is close to the ground truth. Simultaneously, by applying BCE to the reinforcement maps, the model is trained to accurately identify and classify specific key points or features within the image. The combined loss function is expressed in the Equation 3.

$$\mathcal{L}_{\text{combined}} = \alpha \cdot \mathcal{L}_{\text{MAE}} + \beta \cdot \mathcal{L}_{\text{BCE}} \tag{3}$$

where $\mathcal{L}_{\text{combined}}$ is the total loss function, $\mathcal{L}_{\text{MAE}}$ is the Mean Absolute Error loss on the density maps, $\mathcal{L}_{\text{BCE}}$ is the Binary Cross Entropy loss on the reinforcement maps. α and β are weighting factors that determine the contribution of each loss term. We used $\alpha = 1000$ and $\beta = 10$ to place greater emphasis on the density map generation.

3.3. Training and Evaluation Details

For consistency across all datasets, we adhered to the same dataset preparation procedure throughout the model development process.

Density Map Generation We generate density maps following the method described in [2]. We convert the ground truth to density maps by, convolving a Normalized Gaussian kernel over delta function $\delta(x - x_i)$ where x_i is a targeted object. We used a window size of 16 and a fixed spread parameter $\sigma = 4$ of the Gaussian kernel to generate the density maps.

Reinforcement Map Generation For reinforcement maps, we used the same method as density generation but used a larger window size of 32 and spread parameter ($\sigma = 8$). Once the blurred map is generated, we use binary thresholding (th = 0.001) to create a classification map to train the reinforcement branch.

3.3.1 Training Details

For training the model, all images were resized to dimensions of 224x224 pixels, with corresponding head coordi-

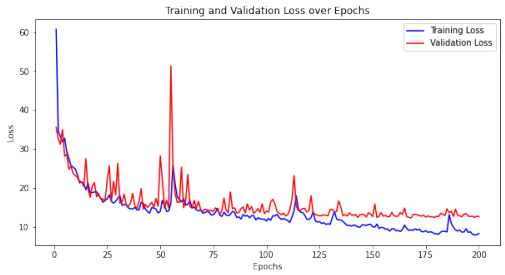

Figure 3. Plot showing the training and validation loss over epochs. The training loss (blue) and validation loss (red) are plotted against the number of epochs.

nate annotations scaled accordingly. Density maps and reinforcement maps were generated based on these resized images as discussed in Section 3.3. Data augmentation was limited to random horizontal flipping with a 50% probability and normalization of the RGB image values. The dataset was split into training and cross-validation datasets as described in Section 3.1.1. The Swin-UNet model was initialized using Kaiming Normal weight initialization to enhance training stability and convergence speed, particularly for the Swin Transformer with GELU activation functions. We employed a combined loss function as described in Section 3.2 and used the Adam optimizer with a learning rate of $1 \times e^{-4}$. We trained the model for 200 epochs, and the resulting training and cross-validation losses across these epochs are shown in Fig. 3.

3.3.2 Evaluation Details

Similar to the training process, all images were resized to 224x224 pixels during evaluation as well. The model's output is processed through a ReLU activation function to eliminate negative values in the predicted density maps. The predicted crowd count and target crowd count are obtained by summing all the values in the generated density maps and the target density maps, respectively. As metrics to evaluate our crowd counting models, we use Mean Absolute error (MAE) and Root Mean Squared Error (RMSE). They are defined as follows:

$$\text{MAE} = \frac{1}{N}\sum_{i=1}^{N}\left|C_i - C_i^{\text{GT}}\right| \tag{4}$$

$$\text{RMSE} = \sqrt{\frac{1}{N}\sum_{i=1}^{N}\left(C_i - C_i^{\text{GT}}\right)^2} \tag{5}$$

where C_i is the predicted count, C_i^{GT} is the ground truth count, and N is the total number of images. MAE is used to indicate how accurate the result is, while RMSE is used to gauge the robustness of the model.

Method	MAE	RMSE
MCNN	277	426
CSRNet	-	-
SANet	-	-
M-SFANet	85.60	151.23
Swin-UNet (Ours)	260.48	260.48

Table 1. Comparison of evaluation metrics on UCF-QNRF dataset.

Method	MAE	RMSE
MCNN	110.2	173.2
CSRNet	68.2	115.0
SANet	67.0	104.5
M-SFANet	59.69	95.66
Swin-UNet (Ours)	140.26	140.26

Table 2. Comparison of evaluation metrics on ShanghaiTech Part-A dataset.

3.4. Compute Resources

Initially, we trained a smaller version of our model using the publicly available $2 \times T4$ GPUs provided by Kaggle. Once we verified that the model was functioning correctly, we developed a deeper model and utilized the cloud computing resources provided by the University at Buffalo's Cloud Computing Research (UB CCR) [11] group for further training and evaluation.

4. Results

In this section, we present the results of crowd counting on 3 datasets, namely, UCF-QNRF [4], ShanghaiTech Part-A [13] and UCF-CC-50 [3] following the approach described in Section 3.3.2.

4.1. UCF-QNRF

We evaluated our model on the 334 images of testing samples from the UCF-QNRF dataset. Samples of the test case are shown in Fig 4. The comparison of the evaluation metrics with the other density map-based approaches us shown in Table 1.

4.2. ShanghaiTech Part-A

We also evaluated our model on the 182 images of testing samples from the ShanghaiTech Part-A dataset. Samples of the test case are shown in Fig 5. The comparison of the evaluation metrics with the other density map-based approaches us shown in Table 2.

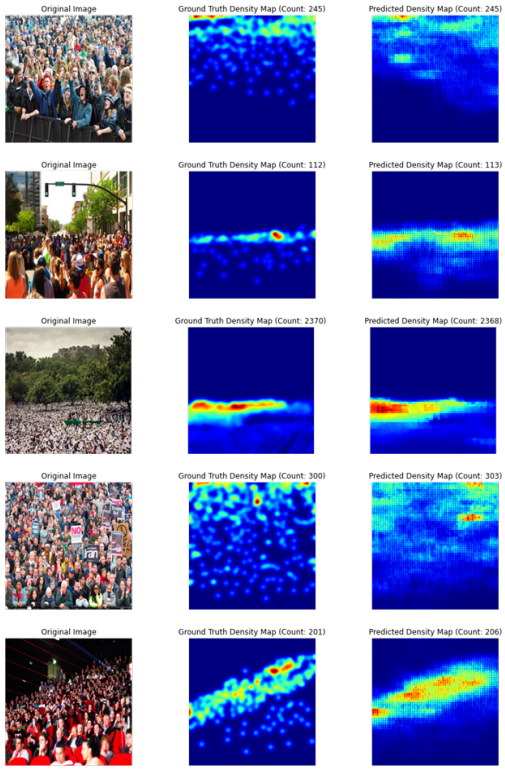

Figure 4. Visualization of estimated density maps. The first column is sample images from UCF-QNRF. The second column is the ground truth density maps with the ground truth crowd count. The third column in the estimated density maps with the estimated crowd count.

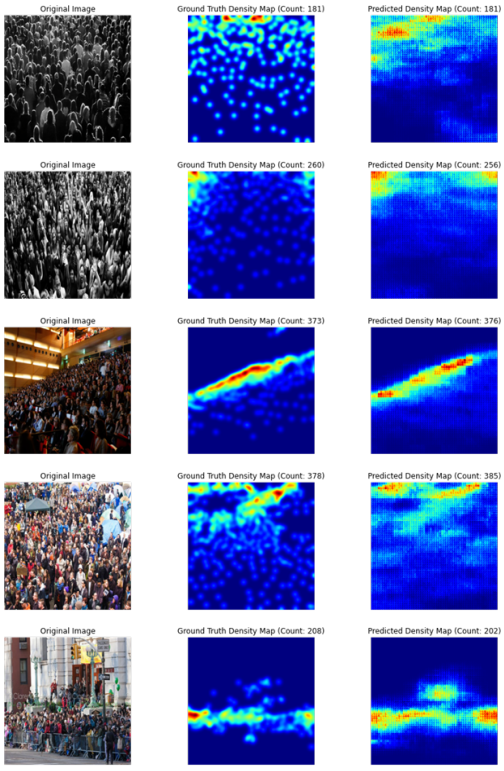

Figure 5. Visualization of estimated density maps. The first column is sample images from ShanghaiTech Part-A. The second column is the ground truth density maps with the ground truth crowd count. The third column in the estimated density maps with the estimated crowd count.

Method	MAE	RMSE
MCNN	377.6	509.1
CSRNet	266.1	397.5
SANet	258.4	334.9
M-SFANet	162.33	276.76
Swin-UNet (Ours)	647.64	647.64

Table 3. Comparison of evaluation metrics on UCF-CC-50 dataset.

4.3. UCF-CC-50

We evaluated our model on this popular benchmark dataset with 50 images. Samples of the test case are shown in Fig 6. The comparison of the evaluation metrics with the other density map-based approaches us shown in Table 3.

5. Analysis

As we see from the visualizations of results in Section 4, our model produces a close approximation of the crowd count. It also yields identical MAE and RMSE values across the datasets. This outcome suggests that the model's errors are consistent in magnitude across predictions, with minimal large deviations. The equivalence of MAE and RMSE indicates that the squared errors do not disproportionately affect the RMSE, implying that most errors are relatively small and uniformly distributed. Consequently, this suggests that the model performs reliably across the dataset, without being significantly impacted by outliers.

However, one limitation of our approach is that the density maps generated by the model lack smoothness as seen in Fig 4, 5, 6. We suspect that this issue arises due to the window-based self-attention mechanism in the underlying Swin Transformer network giving sharp transitions or discontinuities at the boundaries of these windows. To address this, applying Gaussian smoothing to the generated density maps could produce smoother and more refined outputs. Another limitation is the fixed image size of 224x224 resolution. With better computing resources and more time, we could have utilized the original image resolution, enabling our model to extract more detailed information from the features and potentially improve the accuracy of the crowd count estimation.

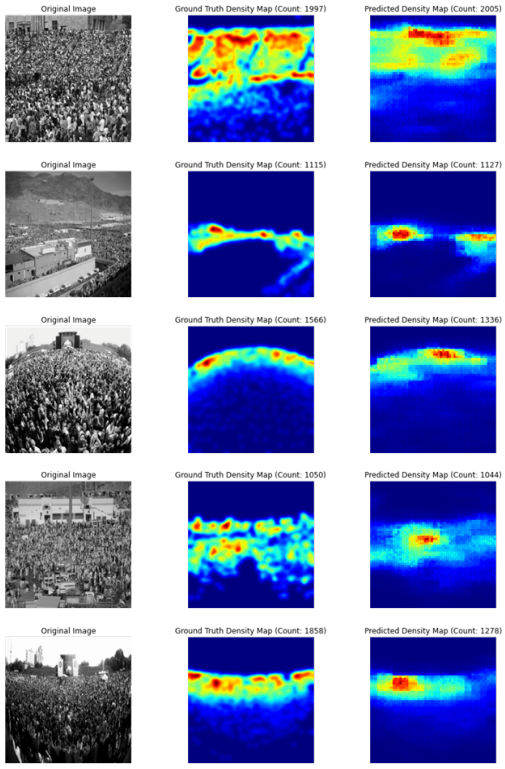

Figure 6. Visualization of estimated density maps. The first column is sample images from UCF-CC-50. The second column is the ground truth density maps with the ground truth crowd count. The third column in the estimated density maps with the estimated crowd count.

6. Discussion and Lessons Learned

Throughout the development of this project, we gained valuable insights into the intricacies of deep learning and its application in solving real-world challenges. Integrating Swin Transformers with the U-Net architecture highlighted the importance of leveraging different model strengths to handle complex tasks, such as accurately counting individuals in densely populated scenes. We also recognized the importance of designing an effective loss function and the role of reinforcement networks in accelerating model convergence. Additionally, we developed expertise in preprocessing and managing diverse datasets, such as ShanghaiTech, UCF-QNRF, and UCF-CC-50, to train a model that generalizes well across various scenarios.

One of the key lessons from this project is the necessity of balancing model complexity with computational efficiency. While the Swin-UNet architecture is powerful, its implementation for dense crowd counting requires careful consideration of resource constraints, especially when scaling to real-time applications.

6.1. Future Work and Extensions

This project could be expanded in multiple ways to increase its scope. Moving towards the recent trends of head point recognition is one area that shows promise for future research. If this feature is improved, the model may be able to estimate crowd counts with even more accuracy—especially when there is severe congestion or people are overlapping. We can also adopt patch-based training and evaluation as discussed in [12], which would enable us to work with higher-resolution images while reducing computational resource requirements.

Furthermore, adding real-time processing and video stream integration would greatly increase the application's usability in dynamic environments. This could involve building a pipeline that processes frames sequentially while preserving temporal consistency or refining the model for faster inference. The accuracy and usability of the model could be further improved by incorporating multi-camera setups for a wider field of view and creating algorithms to manage occlusions and different angles.

References

[1] Hu Cao, Yueyue Wang, Joy Chen, Dongsheng Jiang, Xiaopeng Zhang, Qi Tian, and Manning Wang. Swin-unet: Unet-like pure transformer for medical image segmentation, 2021. 1, 2

[2] Xinkun Cao, Zhipeng Wang, Yanyun Zhao, and Fei Su. Scale aggregation network for accurate and efficient crowd counting. In *Proceedings of the European Conference on Computer Vision (ECCV)*, September 2018. 1, 3

[3] Haroon Idrees, Imran Saleemi, Cody Seibert, and Mubarak Shah. Multi-source multi-scale counting in extremely dense crowd images. In *2013 IEEE Conference on Computer Vision and Pattern Recognition*, pages 2547–2554, 2013. 1, 2, 3, 4

[4] Haroon Idrees, Muhmmad Tayyab, Kishan Athrey, Dong Zhang, Somaya Al-Maadeed, Nasir Rajpoot, and Mubarak Shah. Composition loss for counting, density map estimation and localization in dense crowds, 2018. 1, 2, 4

[5] Yuhong Li, Xiaofan Zhang, and Deming Chen. Csrnet: Dilated convolutional neural networks for understanding the highly congested scenes, 2018. 1

[6] Ze Liu, Yutong Lin, Yue Cao, Han Hu, Yixuan Wei, Zheng Zhang, Stephen Lin, and Baining Guo. Swin transformer: Hierarchical vision transformer using shifted windows, 2021. 1, 2

[7] Hao-Yuan Ma, Li Zhang, and Shuai Shi. Vmambacc: A visual state space model for crowd counting, 2024. 1

[8] Hao-Yuan Ma, Li Zhang, and Xiang-Yi Wei. Fgenet: Fine-grained extraction network for congested crowd counting, 2024. 1

[9] Jihye Ryu and Kwangho Song. Crowd counting and individual localization using pseudo square label. *IEEE Access*, 12:68160–68170, 2024. 1

[10] Pongpisit Thanasutives, Ken-ichi Fukui, Masayuki Numao, and Boonserm Kijsirikul. Encoder-decoder based convolutional neural networks with multi-scale-aware modules for crowd counting. In *2020 25th International Conference on Pattern Recognition (ICPR)*. IEEE, Jan. 2021. 1

[11] University at Buffalo Cloud Computing Research. Cloud computing resources, 2024. Accessed: August 14, 2024. 4

[12] Varun Kannadi Valloli and Kinal Mehta. W-net: Reinforced u-net for density map estimation, 2019. 1, 6

[13] Yingying Zhang, Desen Zhou, Siqin Chen, Shenghua Gao, and Yi Ma. Single-image crowd counting via multi-column convolutional neural network. In *2016 IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*, pages 589–597, 2016. 1, 2, 3, 4