Research Square

Preprints are preliminary reports that have not undergone peer review.
They should not be considered conclusive, used to inform clinical practice,
or referenced by the media as validated information.

HAMNet A Hybrid Attention Multiscale Network for Spatial Crowd Density Estimation

Nitin Kumar Saini

nitinkumarsbl99@gmail.com

Teerthanker Mahaveer University

Mohan Vishal Gupta

Teerthanker Mahaveer University

Ranjana Sharma

SRM Institute of Science and Technology (SRMIST)

Research Article

Keywords:

Posted Date: November 25th, 2026

DOI: https://doi.org/10.21203/rs.3.rs-11300946/v1

License: This work is licensed under a Creative Commons Attribution 4.0 International License.
Read Full License

Additional Declarations: No competing interests reported.

HAMNet A Hybrid Attention Multiscale Network for Spatial Crowd Density Estimation

Nitin Kumar Saini[1*], Mohan Vishal Gupta[2] and Ranjana Sharma[3]

[1,2*]College of Computing Science and Information Technology, Teerthanker Mahaveer University, Moradabad, Uttar Pradesh, India .
[3]Department of Computer Science and Engineering, SRM Institute of Science and Technology (SRMIST), NCR Campus, Ghaziabad, Uttar Pradesh, India .

*Corresponding author(s). E-mail(s): Nitinkumarsbl99@gmail.com;
Contributing authors: Mohan..computers@tmu.ac.in;
ranjanas@srmist.edu.in;

Abstract

Crowd density estimation must preserve spatial detail despite perspective distortion, occlusion, and substantial variation in apparent person scale. HAMNet is a convolutional architecture that combines a VGG16-BN encoder, parallel dilated convolutions, channel-spatial attention, and pixel-shuffle decoding. It uses count-preserving, scale-aware Gaussian density maps derived from bounding-box annotations. The objective combines pixel and structural fidelity with global-count, local-count, and bias constraints. Experiments use 259 crowd images with 42,968 annotations, split into 181 training, 52 validation, and 26 test images. Inference resolution and global count calibration are selected only on the validation split. On the held-out test set, the tuned configuration achieves an MAE of 17.4000, an RMSE of 22.3700, and a MAPE of 10.34%. The stated architecture contains 12.684 million trainable parameters. These results support the feasibility of the formulation on this dataset, but the small single-environment test set and the absence of repeated-seed, matched-baseline, and component-wise experiments do not support claims of statistical superiority or cross-domain generalization.

Keywords: Crowd counting, Density estimation, Spatial analysis, Attention mechanism, Multi-scale feature extraction

1 Introduction

Crowd counting estimates how many people appear in an image. Density estimation extends this task by predicting a spatial density map whose integral yields the count. In addition to the total, the map identifies where people are concentrated, which is useful in public-safety monitoring, urban analysis, and intelligent surveillance [6, 14, 28].

Perspective distortion, occlusion, and background clutter make density estimation difficult. Apparent head size can change substantially within a single image, while overlap can remove much of the local visual evidence. Accurate localization requires fine spatial detail, whereas distinguishing crowd patterns from background texture often requires broader context. Multi-scale architectures address this mismatch in receptive field requirements [3, 8, 15–17], and attention mechanisms can emphasize informative channels and locations [5, 11, 23, 25].

HAMNet combines these mechanisms within a single density-estimation pipeline. It fuses intermediate VGG16-BN features, processes them with parallel dilated convolutions, applies channel and spatial attention sequentially, and uses pixel-shuffle blocks to recover a full-resolution density map. Its objective constrains map fidelity together with image-level and local counts. Because inference scale and count calibration can affect the reported error independently of training, both are selected on the validation split.

1.1 Contributions

The main contributions are as follows:

1. **Hybrid multi-scale representation:** HAMNet combines stride-eight VGG16-BN features, parallel dilated convolutions, and sequential channel and spatial attention.
2. **Count-preserving target construction:** Bounding-box annotations define scale-aware Gaussian targets whose discrete integrals equal the annotated counts.
3. **Joint density and count supervision:** The objective combines pixelwise and structural terms with global-count, multi-scale patch-count, and batch-bias losses.
4. **Validation-controlled inference:** Checkpoint selection, inference scale, optional flip averaging, and count calibration are determined without test labels; the manuscript also documents model complexity and evaluation limitations.

The next section positions HAMNet relative to existing crowd-density methods and identifies the remaining research gap.

2 Related Work

Crowd analysis has progressed from individual-person detection to global count regression and, subsequently, spatial density estimation [1, 6, 28]. Recent work has focused on scale variation, preservation of spatial detail, feature selection, computational efficiency, and difficult imaging conditions.

2.1 Evolution of Crowd Density Estimation

Early detection-based methods searched for individual heads or bodies, an approach that becomes unreliable under heavy occlusion and overlap. Holistic regression avoids resolving each person, but returns only a global count and loses most spatial information. Density-map prediction provides an intermediate formulation: the network estimates a spatial field and obtains the count by integration. Joint density and residual-count supervision showed that these signals can complement one another [28], while head localization adds spatial information [6]. Later work considered settings with distinct constraints. Low-altitude aerial imagery introduces pronounced viewpoint and scale variation [14], whereas resource-constrained applications require lightweight models [1]. Together, these settings expose the limitations of representations tied to a single spatial scale.

2.2 Recent Deep Learning Approaches

Scale-aware CNNs and redesigned multi-scale networks address the fact that one fixed receptive field cannot represent all crowd sizes equally well [3, 17]. Other approaches combine density prediction with object detection [13], choose subnetworks according to estimated crowd level [2], or learn from count-level supervision when dense pixel annotations are unavailable [24]. Model compression and teacher–student transfer have also been used to reduce inference cost [9, 10]. Cross-level attention, transformer representations, region-adaptive objectives, and cascaded scale-driven networks offer further ways to model context and nonuniform crowd distributions [4, 5, 11, 16]. These studies suggest that performance depends not only on backbone capacity, but also on how a method organizes scale and spatial context.

2.3 Attention, Multimodal and Robust Crowd Counting

Attention can operate over channels, spatial locations, or feature hierarchies. Spatially aware and multi-scale fusion methods use it to relate crowd location to changes in apparent scale [8, 11, 15]. In multimodal settings, related mechanisms align and weight RGB, thermal, temporal, or frequency-domain signals, particularly when haze or poor visibility weakens conventional image evidence [7, 23, 25, 29, 30, 33]. Domain-generalization methods instead seek features that remain useful when scene appearance changes [22]. HAMNet uses a single visual modality, so it applies attention to the fused feature tensor rather than across sensors. The channel stage reweights feature responses, and the spatial stage emphasizes locations relevant to density reconstruction.

2.4 Direct Comparison with Closely Related Architectures

The labels multi-scale CNN and attention-based counting do not, by themselves, define HAMNet's technical contribution. Table 1 compares HAMNet with the HMoDE+REL design, which uses multi-resolution density experts and hierarchical gates [3], and with CAMCNet, which uses multi-scale context and cross-level attention [11]. This is a structural comparison, not a ranking across datasets; "NR" marks unavailable details, and "–" indicates that no common test-set score is available.

Table 1 Architecture-level comparison with closely related crowd-density networks. Reported MAE/RMSE values are meaningful only for the dataset named in the same row; they are not a cross-paper ranking. HAMNet is evaluated on the study-specific 26-image test split.

Method	Back-bone	Multi-scale mechanism	Attention mechanism	Density target	Decoder	Loss / count supervision	Evaluation dataset(s)	Reported MAE / RMSE	Params. (M)
HMoDE + REL [3]	VGG16-based encoder	Hierarchical mixture of density experts at multiple resolutions	Pixelwise hierarchical gates with foreground and second-stage gating	Head-point Gaussian density maps; local counts obtained by integration	Multi-resolution density heads followed by upsampling	Density reconstruction, REL local-count supervision, and gate/expert regularization	Shanghai Tech, UCF_CC_50, UCF-QNRF, JHU-CROWD++, and NWPU-Crowd	Shanghai Tech Part A: 54.4 / 87.4	NR
CAMCNet [11]	NR	High-level context enhancement and shallow–deep feature fusion	Cross-level attention for feature fusion	Density-map supervision (kernel construction NR)	Fused multi-level density-regression head	NR	Multiple public crowd benchmarks	– (no common test set)	NR
HAMNet (this study)	VGG16-BN stages B_1–B_4	Stride-eight F_3/F_4 fusion followed by four parallel dilated branches ($r \in \{1, 2, 3, 4\}$)	Sequential CBAM channel then spatial attention on one fused tensor	Box-scale-aware truncated Gaussian, renormalized after truncation	Two-convolution trunk plus three PS_2 stages to full resolution	L_1, MSE, SSIM, global-count, multi-scale patch-count, and batch-bias terms	Study dataset: 259 images; 26-image held-out test split	17.40 / 22.37	12.684

The relevant distinctions are structural rather than categorical. HMoDE constructs and gates several density experts. HAMNet instead fuses encoder features at a common stride, applies parallel dilated convolutions, and refines one *single* feature tensor with sequential channel and spatial attention. CAMCNet uses attention to exchange information across feature levels, whereas HAMNet applies CBAM after same-stride fusion and restores image resolution through three pixel-shuffle stages. HAMNet also uses a box-scale-aware target that is renormalized after truncation and combines image-level, patch-level, and batch-bias count constraints. These choices define the proposed configuration, but they do not show that any component improves accuracy in isolation; that question requires the component-wise protocol in Section 3.10.

2.5 Research Gaps

Existing work establishes the value of multi-scale processing and attention, but it does not replicate the particular HAMNet configuration. The design couples same-stride VGG16-BN feature fusion, four dilated receptive fields, sequential CBAM refinement, a pixel-shuffle full-resolution decoder, and box-aware, count-preserving density supervision. This is a configuration-level distinction rather than a claim that HAMNet is the first method to combine scale and attention. The current experiments assess feasibility on the study dataset; the planned ablations are needed to quantify the contribution of individual design choices.

3 Proposed Methodology

This section describes the dataset and annotations, image preprocessing and density-target construction, HAMNet architecture, training and inference procedures, and evaluation protocol.

The overall pipeline is summarized in Figure 1.

3.1 Dataset Composition and Scope

The dataset contains 259 crowd images with 42,968 person annotations. Images were captured at the College of Computing Science and Information Technology, Teerthanker Mahaveer University, using a mobile phone from different viewing directions. These viewpoints vary apparent crowd scale and spatial layout, although the collection remains limited to one institution. A fixed image-level split contains 181 training images, 52 validation images, and 26 test images. Each bounding box is converted to a point location and a scale value for density-map generation. Table 2 reports split sizes and count statistics. No result in this paper is based on another dataset.

Figure 2 shows the distribution of images and annotations across splits. The training partition contains most images and annotations, while the validation and test partitions have similar mean counts per image. Figure 3 shows representative scenes and illustrates variation in viewpoint, spatial layout, and crowd size.

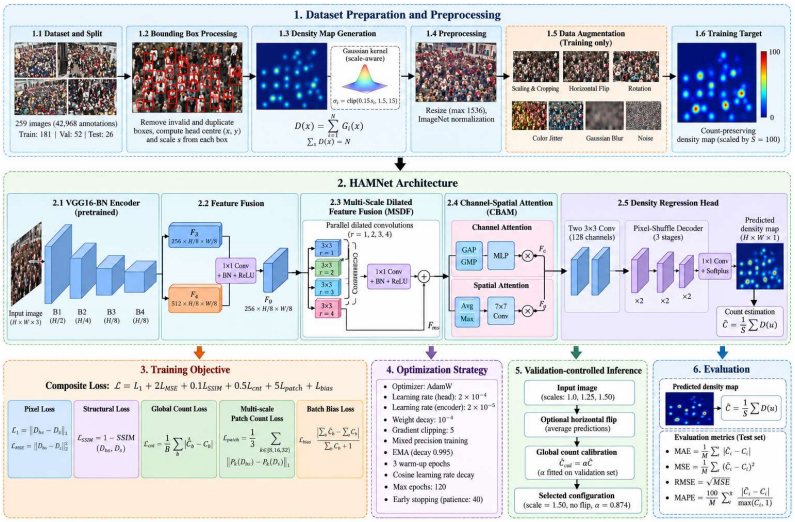
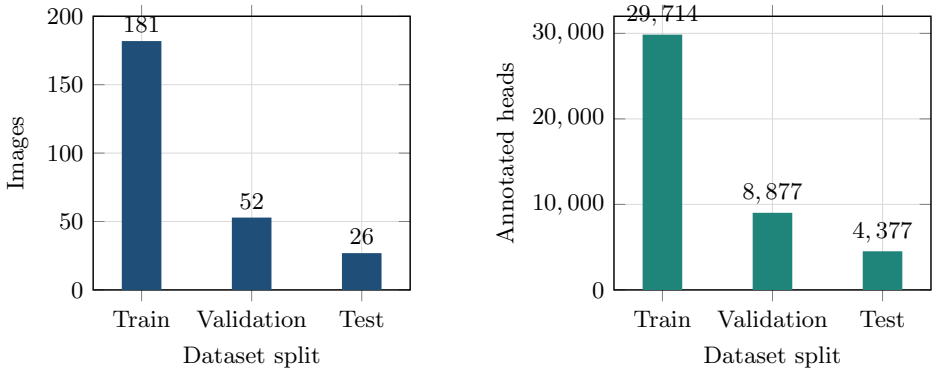


Fig. 1 Overview of the HAMNet methodology, from annotation processing and density-target generation to model training, validation-controlled inference, and evaluation.

Table 2 Summary of the crowd-density-estimation dataset. Mean and standard deviation are reported in annotated heads per image.

Split	Images	Annotations	Min.	Max.	Mean	Std. Dev.
Training	181	29,714	59	281	164.17	34.56
Validation	52	8,877	112	266	170.71	31.84
Testing	26	4,377	122	222	168.35	26.90
All splits	259	42,968	—	—	—	—



(a) Images per split (b) Annotated heads per split

Fig. 2 Dataset composition by split: (a) images and (b) annotated heads.

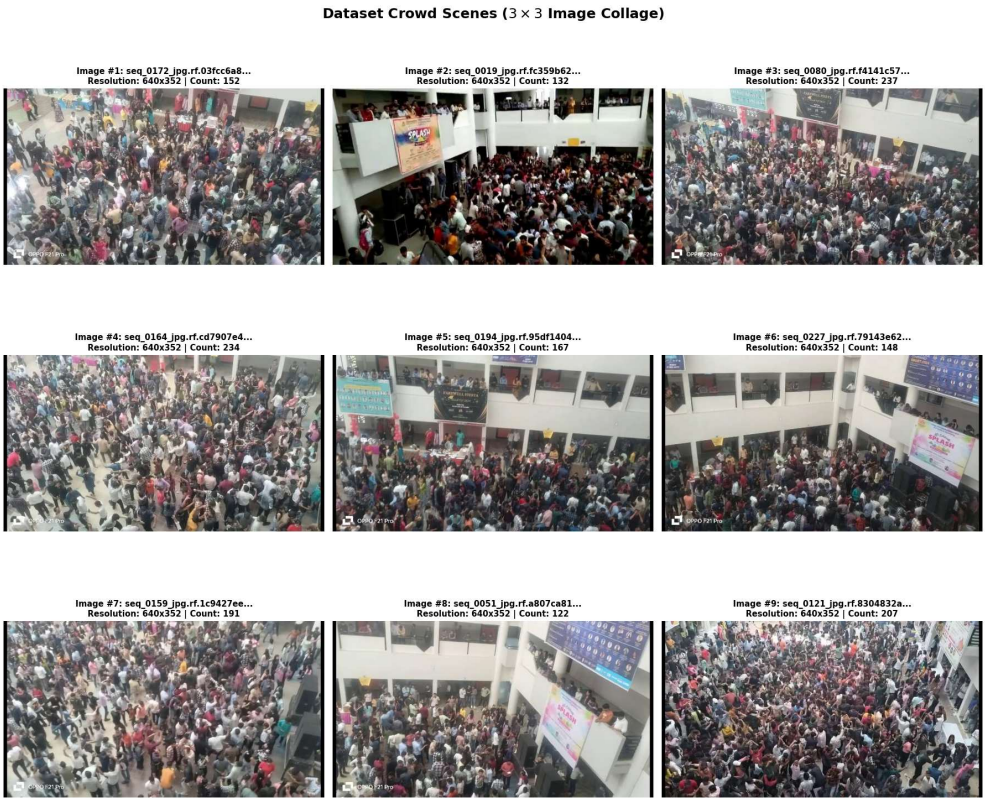


Fig. 3 Representative 3×3 collage of crowd scenes from the dataset. Each panel reports the image resolution and annotated head count, illustrating the variation in viewpoint, spatial distribution, and crowd size within the collection.

3.2 Image Acquisition and Split Integrity

Table 3 summarizes the available acquisition information. Although the mobile-phone images were deliberately captured from different directions, the retained records do not include capture time, event or session, acquisition sequence, device identifier, or scene group. Directional variation therefore does not establish that the images are independent samples.

Because the available annotations cannot determine whether two images came from the same event, session, camera sequence, or near-duplicate scene, related images may appear in different partitions. The split provides filename-level separation, not verified scene-level or sequence-level independence. The present archive does not support reconstruction of a group-based split. If raw capture records or scene and session labels become available, all images from the same group should be assigned to one partition and the study should be re-evaluated under that protocol.

The 26-image test set also limits the precision of comparative conclusions. The study does not include multiple random group splits or repeated-seed training, and

Table 3 Available acquisition and split information.

Item	Available information
Collection site	College of Computing Science and Information Technology, Teerthanker Mahaveer University
Capture device	Mobile phone
Viewpoint	Images taken from different viewing directions
Capture grouping metadata	Event, session, camera/device identifier, date/time, sequence, and scene-group labels are unavailable
Partition	Fixed image-level split: 181 training, 52 validation, and 26 test images; filenames do not overlap

naïve random resampling would not resolve the unobserved dependency issue. The evidence therefore applies only to the reported fixed split. Broader claims require recovered grouping metadata, repeated training across seeds and group-aware splits, and evaluation on at least one established public crowd-counting benchmark.

3.3 Data Annotation Protocol

Annotations are stored as Roboflow-compatible bounding boxes. The preprocessing removes boxes with nonpositive width or height ($x^{\max} \le x^{\min}$ or $y^{\max} \le y^{\min}$) and exact duplicates. Each retained box yields a centre point and a scale descriptor:

$$\mathbf{p}_{j,n} = \left(\frac{x_{j,n}^{\min} + x_{j,n}^{\max}}{2}, \frac{y_{j,n}^{\min} + y_{j,n}^{\max}}{2} \right), \qquad s_{j,n} = \sqrt{(x_{j,n}^{\max} - x_{j,n}^{\min})(y_{j,n}^{\max} - y_{j,n}^{\min})}. \tag{1}$$

For image j, the resulting annotation set is $\mathcal{P}_j = \{(\mathbf{p}_{j,n}, s_{j,n})_{n=1}^{N_j}\}$, where $N_j = |\mathcal{P}_j|$. Figure 4 illustrates the derived centre points. Their coordinates specify the Gaussian-kernel locations, while $s_{j,n}$ determines the local kernel scale.

The total number of annotations is

$$N_{\text{total}} = \sum_{j=1}^{259} N_j = 42{,}968. \tag{2}$$

Geometric preprocessing and augmentation are applied jointly to images, point coordinates, and scale descriptors. The method then rebuilds the density map from the transformed annotations rather than warping a pre-existing target map independently of its points.

Count conservation provides a direct target-construction check. For image j, the discrepancy between the annotation count and the discrete map integral is

$$r_j = \left| N_j - \sum_{\mathbf{u}} D_j(\mathbf{u}) \right|, \tag{3}$$

and a correctly normalized target satisfies $r_j \approx 0$. Equation 6 defines the count-preserving map generated from these point annotations [32].



Fig. 4 Representative crowd images with derived head-centre annotations. Each red marker denotes one retained person annotation used to construct the density target.

3.4 Image Preprocessing and Data Augmentation

Only training images receive stochastic augmentation; validation and test images undergo deterministic preprocessing. Let $I \in [0, 255]^{H \times W \times 3}$ denote an image with annotations $\mathcal{P} = \{(\mathbf{p}_i, s_i)_{i=1}^{N}\}$. If its longest side exceeds 1536 pixels, the image is scaled by $\rho = 1536/\max(H, W)$; otherwise, $\rho = 1$. When the resolution recorded in the annotation CSV differs from the loaded image, coordinates are corrected before resizing. The image and annotations are resized together:

$$I_r = \mathcal{R}_{\rho H, \rho W}(I), \qquad \mathbf{p}_{i,r} = \rho \mathbf{p}_i, \qquad s_{i,r} = \rho s_i, \tag{4}$$

where $\mathcal{R}$ denotes bicubic resampling. Pixel values are mapped to $[0,1]$ and standardized channel-wise with the ImageNet statistics used by the pretrained encoder:

$$\widetilde{I}_{r,c} = \frac{I_{r,c}/255 - \mu_c}{\sigma_c}, \qquad c \in \{R, G, B\}. \tag{5}$$

Density targets are generated at the resized image resolution. In the box-aware formulation, Gaussian width depends on the square root of the transformed box area and is bounded to avoid excessively narrow or diffuse kernels:

$$\begin{aligned} D(\mathbf{x}) &= \sum_{i=1}^{N} G_i(\mathbf{x}), \\ G_i(\mathbf{x}) &= \frac{\exp\!\left(-\|\mathbf{x} - \mathbf{p}_{i,r}\|_2^2/(2\sigma_i^2)\right)}{\sum_{\mathbf{v}\in\mathcal{N}_i} \exp(-\|\mathbf{v} - \mathbf{p}_{i,r}\|_2^2/(2\sigma_i^2))}\, \mathbb{I}[\mathbf{x} \in \mathcal{N}_i], \\ \sigma_i &= \mathrm{clip}(0.15 s_{i,r}, 1.5, 15), \qquad \sum_{\mathbf{x}} G_i(\mathbf{x}) = 1, \end{aligned} \tag{6}$$

Each G_i is truncated to a $3\sigma_i$ neighbourhood and then renormalized. Consequently, $\sum_{\mathbf{x}} D(\mathbf{x}) = N$, so the discrete map retains the image-level count. Conventional geometry-adaptive targets often derive bandwidth from neighbouring annotations [32]; here, the observed box scale determines the kernel width.

3.5 Density-Target Visualization

Figure 5 presents three representative ground-truth density maps generated by Equation 6. After normalization, each Gaussian contributes one person, so the discrete integral of each map equals the annotated count shown above it. Each panel uses its own color scale and should be interpreted accordingly. The figure illustrates supervision-target construction, not model predictions.

With probability 0.7, each training image is rescaled by $a \sim \mathcal{U}(0.85, 1.50)$ before extracting a 512×512 crop. Half of these crops are centred near a randomly selected annotation to reduce empty samples. Horizontal reflection is applied with probability 0.5, and rotation by $\phi \sim \mathcal{U}(-8^\circ, 8^\circ)$ is applied with probability 0.25. Under the geometric transform $T(\mathbf{x}) = A\mathbf{x} + \mathbf{b}$, the image and annotations become

$$\begin{aligned} I_T(\mathbf{u}) &= I_r\!\left(T^{-1}(\mathbf{u})\right), \\ \mathbf{p}_{i,T} &= A\mathbf{p}_{i,r} + \mathbf{b}, \qquad s_{i,T} = \sqrt{|\det A|}\, s_{i,r}. \end{aligned} \tag{7}$$

Annotations falling outside the transformed crop are discarded, and the density target is rebuilt from the remaining points. Photometric augmentation includes Gaussian blur ($p = 0.1$), grayscale conversion ($p = 0.08$), brightness, contrast, and saturation changes bounded by ± 0.25, gamma correction with $\gamma \sim \mathcal{U}(0.8, 1.25)$ ($p = 0.3$), and additive Gaussian noise with standard deviation 0.02 ($p = 0.2$). These operations follow established augmentation practice [19, 20]. Validation and test samples receive no stochastic augmentation.

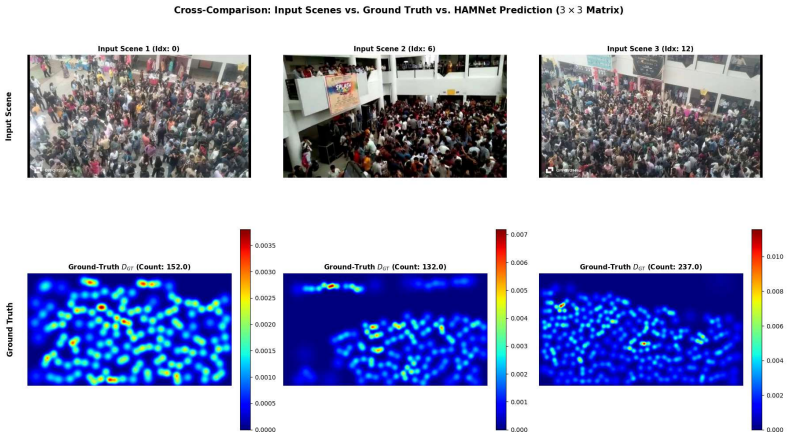


Fig. 5 Examples of count-preserving ground-truth density maps generated from box-derived Gaussian kernels. Brighter regions indicate higher local density; each map integrates to its corresponding annotated count.

3.6 HAMNet Architecture

For a normalized image $\widetilde{I}$, HAMNet v3 predicts a non-negative scaled density map $\widehat{D}_s$, which is integrated to obtain the crowd count. The encoder uses the first four VGG16-BN stages [21], denoted by B_1–B_4. It concatenates two stride-eight tensors, F_3 and F_4, and projects them to 256 channels:

$$F_3 = B_3(B_2(B_1(\widetilde{I}))), \qquad F_4 = B_4(F_3), \qquad F_0 = \mathrm{ReLU}(\mathrm{BN}(W_{1\times1} * [F_3 \,\|\, F_4])) \,. \quad (8)$$

The tensors have dimensions $F_3 \in \mathbb{R}^{256\times H/8\times W/8}$ and $F_4 \in \mathbb{R}^{512\times H/8\times W/8}$. Their concatenation contains 768 channels, which the 1×1 layer reduces to 256. During fine-tuning, the running statistics of the pretrained batch-normalization layers remain fixed.

Multi-scale dilated fusion (MSDF) applies four parallel 3×3 convolutions with dilation factors $\mathcal{R} = \{1, 2, 3, 4\}$. Their outputs are concatenated, fused, and added to the input through a residual connection:

$$F_{\mathrm{ms}} = F_0 + \mathrm{ReLU}\left(\mathrm{BN}\left(W_{1\times1} * \left[\mathrm{Conv}^{(r)}_{3\times3}(F_0)\right]_{r\in\mathcal{R}}\right)\right). \quad (9)$$

Dilated kernels expand the receptive field without reducing feature-map resolution [31]. A convolutional block attention module (CBAM) then refines the fused tensor [27]. Let GAP and GMP denote global average and maximum pooling, respectively, and

let g denote the shared channel MLP. Channel and spatial attention are computed as

$$\begin{aligned} A_c &= \mathrm{sigmoid}(g(\mathrm{GAP}(F_{\mathrm{ms}})) + g(\mathrm{GMP}(F_{\mathrm{ms}}))\,, \\ F_c &= A_c \odot F_{\mathrm{ms}}, \\ A_s &= \mathrm{sigmoid}(W_{7\times7} * [\mathrm{Mean}_c(F_c) \parallel \mathrm{Max}_c(F_c)])\,, \\ F_a &= A_s \odot F_c. \end{aligned} \tag{10}$$

The shared MLP uses a $256 \to 16 \to 256$ bottleneck, and a 7×7 convolution produces the spatial-attention map.

The density-regression head first forms a 128-channel representation with two 3×3 convolutions and then applies three pixel-shuffle upsampling stages. For $Z_0 = \mathrm{Trunk}(F_a)$, stage ℓ is

$$Z_{\ell+1} = \mathrm{ReLU}\Big(\mathrm{PS}_2\Big(W_{3\times3}^{(\ell)} * Z_\ell\Big)\Big)\,, \quad W_{3\times3}^{(\ell)} : 128 \to 512, \quad \ell = 0, 1, 2, \tag{11}$$

where PS_2 doubles the height and width while restoring 128 output channels [18]. The final density map and count are

$$\widehat{D}_s = \mathrm{softplus}(W_{1\times1} * Z_3), \qquad \widehat{C} = \frac{1}{S}\sum_{\mathbf{u}} \widehat{D}_s(\mathbf{u}), \qquad S = 100. \tag{12}$$

Softplus ensures non-negative density values. The constant S rescales the training targets and is divided out when integrating the density map.

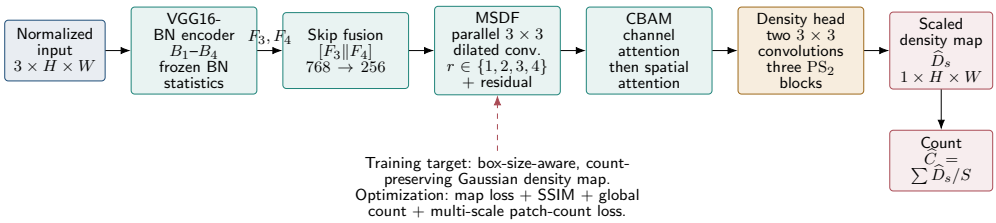


Fig. 6 HAMNet v3 architecture. A VGG16-BN encoder supplies stride-eight features that are fused, processed by multi-scale dilated convolution and CBAM, and decoded to full resolution by three pixel-shuffle blocks. The density-map integral yields the predicted count.

3.7 Model Complexity

Table 4 gives an analytical parameter breakdown derived from the layer specification. The calculation includes convolutional biases and affine batch-normalization parameters, four $256 \to 256$ MSDF branches, the shared CBAM multilayer perceptron in Equation 10, and the complete regression head. VGG convolutional weights and affine batch-normalization parameters remain trainable, although they use the lower encoder

learning rate. Running means and variances are excluded because they are buffers. Under these assumptions, HAMNet has 12,683,572 trainable parameters.

Table 4 Analytical parameter count for the architecture specified in Section 3.6.

Module	Parameters
VGG16-BN stages B_1–B_4	7,640,640
Stride-eight skip projection	197,376
MSDF branches and fusion	2,623,232
CBAM channel and spatial attention	8,563
Two-convolution density trunk	442,624
Three pixel-shuffle convolutions	1,771,008
Final density projection	129
Total trainable parameters	12,683,572

The value in Table 4 is derived from the architecture rather than an executable model summary. No fixed-resolution profiling record is available for FLOPs, activation memory, or latency; consequently, the paper does not make claims about deployment efficiency.

3.8 Training Objective and Inference Protocol

Before optimization, the count-preserving target is multiplied by $S = 100$ to increase gradient magnitude. The objective combines pixelwise reconstruction, structural similarity, global count, local patch count, and batch-level bias:

$$\mathcal{L} = \mathcal{L}_1 + 2\mathcal{L}_{\text{MSE}} + 0.1\mathcal{L}_{\text{SSIM}} + 0.5\mathcal{L}_{\text{cnt}} + 5\mathcal{L}_{\text{patch}} + \mathcal{L}_{\text{bias}}, \tag{13}$$

Here, $\mathcal{L}_1 = \|\widehat{D}_s - D_s\|_1$, $\mathcal{L}_{\text{MSE}} = \|\widehat{D}_s - D_s\|_2^2$, and $\mathcal{L}_{\text{SSIM}} = 1 - \text{SSIM}(\widehat{D}_s, D_s)$. SSIM uses an 11×11 Gaussian window with standard deviation 1.5 [26]. The remaining terms are defined as

$$\begin{aligned}
\mathcal{L}_{\text{cnt}} &= \frac{1}{B}\sum_{b=1}^{B} |\widehat{C}_b - C_b|, \\
\mathcal{L}_{\text{bias}} &= \frac{\left|\sum_{b=1}^{B} \widehat{C}_b - \sum_{b=1}^{B} C_b\right|}{\sum_{b=1}^{B} C_b + 1}, \\
\mathcal{L}_{\text{patch}} &= \frac{1}{3}\sum_{k\in\{8,16,32\}} \left\| \frac{k^2}{S}\,\text{AvgPool}_k(\widehat{D}_s) - \frac{k^2}{S}\,\text{AvgPool}_k(D_s) \right\|_1 .
\end{aligned} \tag{14}$$

The patch term compares local counts at three pooling scales in addition to the image-level constraint. All coefficients in Equation 13 remain fixed during the reported run, so

the experiment does not estimate their individual effects or sensitivity. Section 3.10.1 defines the minimum experiment needed to support claims about the proposed objective. Training uses AdamW [12]. The density-estimation head uses a learning rate of 2×10^{-4}, and the VGG blocks use one tenth of that value. Weight decay is 10^{-4}, gradient norms are clipped at 5, and mixed-precision training is used. An exponential moving average of the weights is maintained with decay 0.995. The learning rate increases linearly for three epochs before following cosine decay:

$$\eta_t = \begin{cases} \eta_0 (t+1)/T_w, & t < T_w, \\ \eta_0 \left[0.01 + 0.99 \, \dfrac{1 + \cos(\pi q_t)}{2} \right], & t \geq T_w, \end{cases} \qquad q_t = \frac{t - T_w}{T - T_w}. \tag{15}$$

Training runs for at most 120 epochs and stops when validation MAE has not improved for 40 epochs.

When horizontal-flip test-time augmentation is enabled, predictions for the original and reflected images are aligned and averaged:

$$\widehat{D}_{\text{tta}} = \frac{\widehat{D}(I) + \text{Flip}^{-1}(\widehat{D}(\text{Flip}(I)))}{2}, \qquad \widehat{C}_{\text{tta}} = \frac{\widehat{C}(I) + \widehat{C}(\text{Flip}(I))}{2}. \tag{16}$$

The validation search considers input scales $\{1.00, 1.25, 1.50\}$, flip averaging on or off, and optional count calibration. When calibration is used, $\alpha = \sum_i C_i \widehat{C}_i / \sum_i \widehat{C}_i^2$ is fitted through the origin using validation predictions, yielding the adjusted count $\alpha \widehat{C}_i$. Held-out evaluation reports MAE, MSE, RMSE, and MAPE separately.

3.9 Implementation and Reproducibility Record

The available experiment record distinguishes retained training choices from essential information that was not retained. The reported single run cannot be reproduced exactly because the execution environment and several run-level settings are unknown. These details are reported as unavailable rather than reconstructed after the fact.

Before submitting or reusing the reported checkpoint, the authors should archive a machine-readable training configuration, source-code revision identifier, checkpoint checksum, package lockfile, and system report containing CPU/GPU, driver, CUDA, cuDNN, Python, and PyTorch versions. The run configuration should also record batch size, worker count, all random seeds, and deterministic and cuDNN benchmark settings. A supplementary reproducibility file should link these materials to the checkpoint used for the reported test results. Until such an archive is available, the result should be treated as a documented single training outcome rather than an exactly reproducible run.

3.10 Component-Wise Ablation Protocol

The complete HAMNet configuration contains several architectural, supervisory, and inference choices. One score for the full system cannot show whether an observed error change arises from MSDF, attention, the decoder, the density target, a loss

term, inference resolution, calibration, or their interaction. Table 5 therefore defines the sequential architecture–loss ablation needed to attribute these effects. Every configuration must use the same fixed split, initialization policy, optimizer, learning-rate schedule, training budget, early-stopping rule, and validation-only model-selection procedure. The A_0 baseline uses the VGG16-BN encoder with a direct density-regression head and bilinear upsampling; all rows retain the pixelwise and SSIM map terms.

Table 5 Pre-specified sequential ablation for the HAMNet architecture and loss terms. CA and SA denote channel and spatial attention, respectively; PS denotes the pixel-shuffle decoder; L_{cnt}, L_{patch}, and L_{bias} are the global-count, local patch-count, and batch-bias terms.

Configuration	MSDF	CA	SA	PS	L_{cnt}	L_{patch}	L_{bias}
A_0: VGG16-BN baseline	×	×	×	×	×	×	×
A_1: A_0 + MSDF	✓	×	×	×	×	×	×
A_2: A_1 + channel attention	✓	✓	×	×	×	×	×
A_3: A_2 + spatial attention	✓	✓	✓	×	×	×	×
A_4: A_3 + pixel shuffle	✓	✓	✓	✓	×	×	×
A_5: A_4 + global-count loss	✓	✓	✓	✓	✓	×	×
A_6: A_5 + patch-count loss	✓	✓	✓	✓	✓	✓	×
A_7: full HAMNet	✓	✓	✓	✓	✓	✓	✓

The density-target ablation should compare the current box-aware truncated Gaussian target, which is renormalized after truncation, against an otherwise identical target without renormalization. This contrast tests count preservation while holding box-derived kernel widths fixed. Inference choices require a separate 2×2 evaluation of scale $s \in \{1.00, 1.50\}$ and calibration on or off. For each scale, the calibrated condition must fit α only on validation predictions; the current selected setting uses $s = 1.50$ and $\alpha = 0.874$. Each configuration should be trained with at least three independent seeds and reported as mean ± standard deviation for MAE and RMSE, together with paired per-image comparisons against the immediately preceding configuration.

3.10.1 Loss-Term Ablation and Coefficient Sensitivity

The cumulative rows A_5–A_7 in Table 5 show how the full architecture can be assembled sequentially, but they do not isolate the effect of each count-related term. The minimum loss experiment in Table 6 therefore fixes the full HAMNet architecture, density target, optimizer, training budget, and data split, while changing only the objective. Let

$$\mathcal{L}_{\text{map}} = \mathcal{L}_1 + 2\mathcal{L}_{\text{MSE}} + 0.1\mathcal{L}_{\text{SSIM}}. \tag{17}$$

The first four rows evaluate the map-only baseline and each count-related term independently; the final row evaluates their joint use.

Each row should be trained with at least three independent seeds. The primary comparison should use one common, pre-specified inference protocol without post-hoc test-set calibration or resolution selection, so that inference changes are not mistaken for loss effects. MAE and RMSE should be reported for every seed, alongside mean

Table 6 Minimum loss-function ablation required for the proposed objective. All configurations use the complete HAMNet architecture and the same count-preserving target.

Configuration	$\mathcal{L}_{\text{cnt}}$	$\mathcal{L}_{\text{patch}}$	$\mathcal{L}_{\text{bias}}$	Objective
L_0: map baseline	×	×	×	$\mathcal{L}_{\text{map}}$
L_{cnt}: global count	✓	×	×	$\mathcal{L}_{\text{map}} + 0.5\mathcal{L}_{\text{cnt}}$
L_{patch}: local patches	×	✓	×	$\mathcal{L}_{\text{map}} + 5\mathcal{L}_{\text{patch}}$
L_{bias}: batch bias	×	×	✓	$\mathcal{L}_{\text{map}} + \mathcal{L}_{\text{bias}}$
L_{full}: proposed loss	✓	✓	✓	Equation 13

± standard deviation and paired per-image error differences relative to L_0. A complementary leave-one-out analysis, in which $\mathcal{L}_{\text{full}}$ is evaluated after removing one of $\mathcal{L}_{\text{cnt}}$, $\mathcal{L}_{\text{patch}}$, or $\mathcal{L}_{\text{bias}}$, would further quantify the conditional role of each term.

Coefficient sensitivity should then vary one major coefficient at a time, keeping the full architecture and all other coefficients fixed: $\lambda_{\text{cnt}} \in \{0.25, 0.50, 1.00\}$, $\lambda_{\text{patch}} \in \{2.5, 5, 10\}$, and $\lambda_{\text{bias}} \in \{0.5, 1, 2\}$. The current coefficients, $(0.5, 5, 1)$, lie at the centre of these ranges. This protocol would show whether the reported configuration is stable under moderate changes to the count-related weights. Until it is completed, the coefficients should be regarded as fixed design settings rather than empirically optimized values.

3.11 Evaluation Metrics

For an evaluation set of M images, let C_i and $\widehat{C}_i$ denote the reference and predicted counts for image i. Mean absolute error (MAE) is

$$\text{MAE} = \frac{1}{M} \sum_{i=1}^{M} \left| \widehat{C}_i - C_i \right|. \tag{18}$$

Mean squared error (MSE) and its square root, RMSE, assign greater weight to large deviations:

$$\text{MSE} = \frac{1}{M} \sum_{i=1}^{M} \left(\widehat{C}_i - C_i \right)^2, \qquad \text{RMSE} = \sqrt{\text{MSE}}. \tag{19}$$

Mean absolute percentage error (MAPE) normalizes each absolute error by the corresponding count; $\max(C_i, 1)$ prevents division by zero. The tables also report its arithmetic complement, $100 - \text{MAPE}$:

$$\text{MAPE} = \frac{100}{M} \sum_{i=1}^{M} \frac{\left| \widehat{C}_i - C_i \right|}{\max(C_i, 1)}, \qquad \text{Score}_{100-\text{MAPE}} = 100 - \text{MAPE}. \tag{20}$$

Lower MAE, RMSE, and MAPE indicate lower count error. The $100-\text{MAPE}$ quantity is reported only as a descriptive complement and is not a classification accuracy.

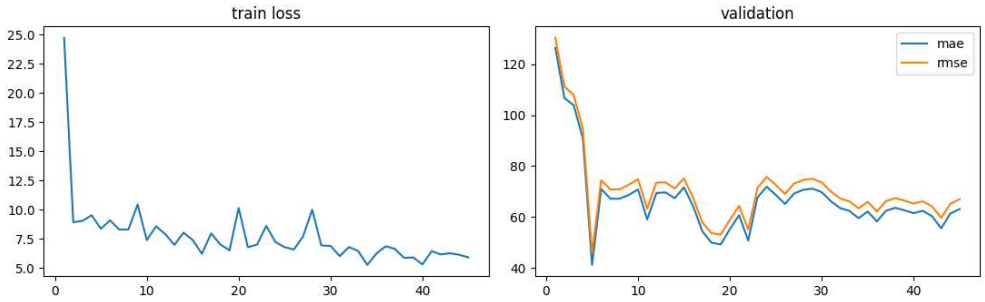


Fig. 7 Training composite loss and validation count errors across epochs. The panels use different vertical scales because the training objective and validation metrics are different quantities.

4 Results and Discussion

4.1 Training and Model Selection

Training stopped at epoch 45 under the 40-epoch early-stopping criterion. At native resolution, validation performance was best at epoch 5, with an MAE of 41.22 and an RMSE of 45.86. The epoch-5 weights were therefore used for the inference-configuration search.

Only validation data informed the choice of input scale, horizontal-flip averaging, and count calibration. The selected configuration used scale 1.50, no flip averaging, and $\alpha = 0.874$. Table 7 compares its validation performance with native-scale inference. Neither checkpoint selection nor inference tuning used test labels.

Table 7 Validation performance of the epoch-5 checkpoint under native and selected inference configurations.

Inference configuration	MAE	RMSE	$100 - \text{MAPE}$ (%)
Native scale 1.00, no flip, no calibration	41.22	45.86	75.9
Selected scale 1.50, no flip, $\alpha = 0.874$	15.86	20.30	90.5

4.2 Training Dynamics

Figure 7 shows the optimization history. Although the training objective generally decreased, validation MAE reached its minimum at epoch 5 and then varied. Using the final epoch would therefore have selected a checkpoint with poorer validation count error. The panels show temporal trends rather than directly comparable values: one reports the composite loss in Equation 13, and the other reports count-based validation metrics.

4.3 Held-Out Test Performance

Table 8 reports held-out performance for the tuned HAMNet configuration. The 26 test images have a mean ground-truth count of 168.35. HAMNet achieves an MAE of 17.4000, an MSE of 500.3000, an RMSE of 22.3700, and a MAPE of 10.34%. The arithmetic complement of MAPE, reported as Average Counting Accuracy, is 89.66%. MAE and RMSE remain the primary count-error measures because this accuracy value is derived from MAPE rather than a classification criterion.

Table 8 Accuracy performance of the tuned HAMNet model configuration as a hold-out dataset. MSE refers to mean square error while the last percentage column represents the complement of MAPE.

MAE	MSE	RMSE	MAPE (%)	100 − MAPE (%)	Test images
17.4000	500.3000	22.3700	10.34	89.66	26

4.4 Per-Image Density and Count Calculation

The aggregate values in Table 8 are computed from a density map and count for each test image. For image i, let $\widehat{D}_{s,i}(\mathbf{u})$ denote HAMNet's scaled output at pixel location $\mathbf{u}$ in image domain Ω_i. Because the training targets are scaled by $S = 100$, the density map in count-preserving units is recovered as

$$\widehat{D}_i(\mathbf{u}) = \frac{\widehat{D}_{s,i}(\mathbf{u})}{S}, \qquad \mathbf{u} \in \Omega_i, \qquad S = 100. \tag{21}$$

Each map value represents an estimated count contribution at its pixel, so integrating the map yields the image-level crowd count. The uncalibrated and calibrated counts are

$$\widetilde{C}_i = \sum_{\mathbf{u}\in\Omega_i} \widehat{D}_i(\mathbf{u}), \qquad \widehat{C}_i = \alpha \widetilde{C}_i, \qquad \alpha = 0.874, \tag{22}$$

where α is the global coefficient selected on the validation split. Without calibration, $\alpha = 1$. For an image with height H_i and width W_i, the image-wide mean predicted density is

$$\overline{\rho}_i = \frac{\widehat{C}_i}{H_i W_i} \quad \text{persons per pixel.} \tag{23}$$

The same map can be integrated over any region $R \subseteq \Omega_i$ to estimate the number of people in that part of the scene:

$$\widehat{C}_i(R) = \alpha \sum_{\mathbf{u}\in R} \widehat{D}_i(\mathbf{u}). \tag{24}$$

For each image, the corresponding count error is $e_i = \widehat{C}_i - C_i$, with absolute and percentage errors given by

$$\mathrm{AE}_i = |e_i|, \qquad \mathrm{APE}_i = 100\frac{|e_i|}{\max(C_i, 1)}. \tag{25}$$

The MAE, MSE, RMSE, and MAPE in Table 8 summarize these per-image quantities over the 26 held-out images. The density map identifies where people are concentrated, and its integral provides the estimated total count.

4.5 Comparison with Benchmark Models

Table 9 compares the tuned HAMNet configuration with five convolutional benchmarks on the study test split. HAMNet has the lowest reported MSE (500.3) and RMSE (22.37), whereas VGG16 has the lowest reported MAE (17.38) and MAPE (9.86%). The MAE difference between VGG16 and HAMNet is 0.02 people. Figure 8 displays the MAE and RMSE values on a shared count-error scale.

Table 9 Comparison between HAMNet and benchmark algorithms using the test split from the experiment. Accuracy is defined by 100 − MAPE. A dash means that the corresponding HAMNet time is not reported in the provided benchmark data.

Model	Params (M)	MAE	MSE	RMSE	MAPE (%)	Accuracy (%)	ms/image	Train (min)
VGG16	9.19	17.38	505.7	22.49	9.86	90.14	22.2	11.2
HAMNet	12.68	17.40	500.3	22.37	10.34	89.66	—	—
AlexNet	3.43	18.22	503.7	22.44	10.73	89.27	10.2	6.4
ResNet50	11.27	18.63	547.0	23.39	10.83	89.17	16.6	10.7
CSRNet	16.26	21.13	702.5	26.50	12.75	87.25	20.6	4.3
MCNN	0.13	33.40	1993.6	44.65	19.80	80.20	15.9	3.4

The supplied VGG16 training history reports a best validation MAE of 19.6975. Horizontal-flip test-time augmentation increased its MAE to 21.7329, MSE to 699.3991, and RMSE to 26.4462; Table 9 therefore uses the non-TTA VGG16 model. A complete matched baseline protocol and per-image predictions are still needed for statistical comparison.

4.6 Sensitivity to Inference Configuration

Table 10 compares the two evaluated inference settings. Moving from native-scale inference without calibration to scale 1.50 with calibration reduced MAE by 13.33 and RMSE by 14.31. Because scale and calibration changed together, this comparison cannot attribute the reduction to either factor alone.

4.7 Ablation Status and Required Evidence

The current experimental record contains only the final HAMNet run. It does not include the component-wise training runs in Table 5 or the five loss configurations in Table 6. Consequently, the held-out MAE of 17.4000 cannot be assigned to MSDF,

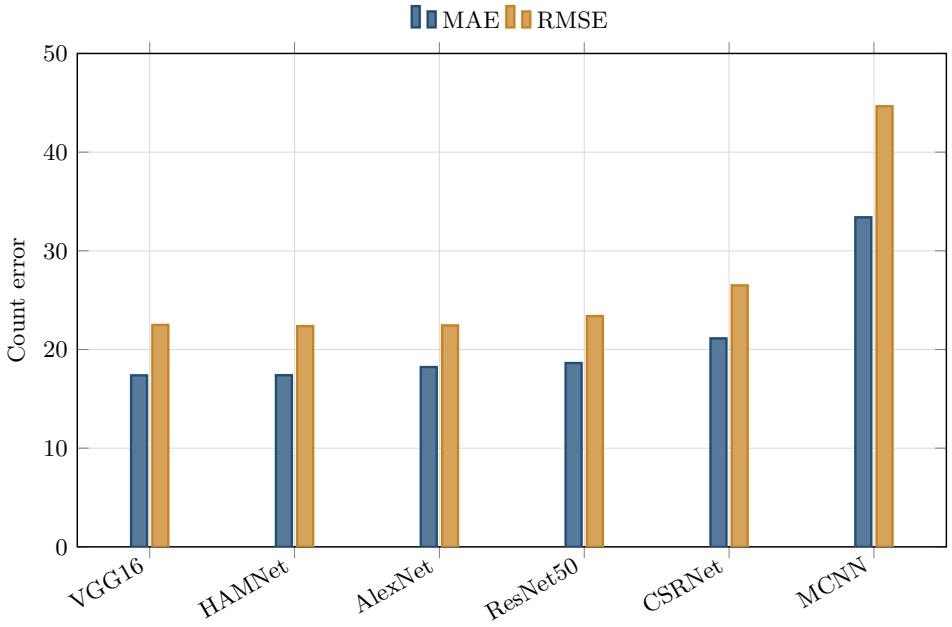


Fig. 8 MAE and RMSE comparison on the study test split. Lower bars indicate smaller count errors. The figure is descriptive because repeated runs, confidence intervals, and paired predictions are unavailable.

Table 10 Held-out performance under the two evaluated inference configurations.

Input scale	Calibration	MAE	RMSE
1.00	No	30.73	36.68
1.50	Yes ($\alpha = 0.874$)	17.4000	22.3700

channel attention, spatial attention, pixel-shuffle decoding, the count-preserving target, an individual loss term, calibration, or increased inference resolution. In particular, the coefficients $(0.5, 5, 1)$ in Equation 13 have not been subjected to sensitivity analysis and should be treated as fixed settings rather than validated optima. The two settings in Table 10 also change scale and calibration simultaneously, so they do not isolate either effect.

Component-level claims require the sequential ablation, target comparison, loss-term and coefficient-sensitivity study in Section 3.10.1, and the full scale–calibration 2×2 evaluation described in Section 3.10. For each configuration, the report should include validation-selected checkpoint details, test MAE and RMSE for every seed, mean $\pm$ standard deviation, per-image predictions, and paired uncertainty intervals relative to the preceding configuration. Until then, the proposed components should

be interpreted as a jointly evaluated design rather than independently validated contributions.

4.8 Discussion

The architecture combines multiple receptive-field scales, attention-based feature refinement, full-resolution decoding, and count-aware supervision. The aggregate result, however, does not establish the causal contribution of any one component. The available record cannot determine whether the observed error reflects MSDF, CBAM, pixel-shuffle decoding, count-preserving target generation, the count-related loss terms, inference resolution, calibration, or their interaction. The component-wise protocol in Table 5 is therefore necessary before any design choice is described as empirically beneficial.

Table 9 offers a descriptive comparison with the supplied benchmark values. VGG16 has a slightly lower MAE and MAPE, whereas HAMNet has the lowest reported MSE and RMSE. Because the available aggregate values lack repeated seeds, paired predictions, and a matched baseline protocol, they do not justify a claim of statistical superiority.

Post-training decisions are part of the evaluation protocol. Higher input resolution may retain evidence for smaller heads, whereas a scalar calibration factor directly adjusts systematic count bias. Since scale and calibration change together in the available record, their separate effects are unknown. Future comparisons should apply the same validation-only selection procedure to every model.

The divergence between training and validation trajectories has two plausible explanations: increasing overfitting or a late-stage misalignment between the composite density loss and image-level count error. The available run cannot distinguish these possibilities. It nevertheless supports early stopping and joint reporting of optimization losses and evaluation metrics.

4.9 Limitations

Several factors limit the interpretation of these findings. The test split contains 26 images from one collection environment, and the available metadata do not establish independence at the camera, event, or sequence level. A single training run produced the reported checkpoint, leaving seed-to-seed variation and confidence intervals unknown. Although the manuscript reports descriptive baseline values, complete matched protocols and paired prediction archives are unavailable. Component-wise and loss-term ablations, including coefficient sensitivity analysis, are also absent. As a result, the contributions of MSDF, channel attention, spatial attention, pixel-shuffle decoding, count-preserving target construction, individual loss terms and weights, inference scale, and calibration cannot be separated.

As documented in Section 3.9, the batch size, random seeds, worker count, hardware and software environment, determinism settings, and checkpoint-linked configuration were not retained. Together with missing per-image predictions and qualitative maps, these omissions prevent exact reproduction of the reported run from the present record. The analytical count in Table 4 also requires verification against

the implemented model, and FLOPs, activation memory, and controlled latency have not been measured. The evidence therefore supports conclusions about the reported split only, not claims of statistical superiority, real-time operation, or cross-domain generalization.

5 Conclusion

HAMNet combines stride-eight multi-scale fusion, channel-spatial attention, pixel-shuffle decoding, and count-aware supervision for density estimation. Using inference settings selected on the validation subset, the optimized configuration achieved an MAE of 17.4000, an RMSE of 22.3700, a MAPE of 10.34%, and a derived $100-\text{MAPE}$ score of 89.66% on the 26-image test split. In the descriptive comparison, HAMNet is close to VGG16 and AlexNet in MAE and has the lowest reported MSE and RMSE.

These results establish feasibility on the study split, not the independent value of each design choice. Before publication, the sequential component ablation, count-preserving-target comparison, and scale–calibration 2×2 evaluation in Section 3.10 should be completed with repeated seeds and paired per-image analysis. Broader comparative or generalization claims also require the checkpoint-linked reproducibility archive specified in Section 3.9, matched baseline retraining, implementation-derived complexity measurements, checkpoint-linked qualitative predictions, and evaluation on established cross-scene benchmarks.

Statements and Declarations

Funding

Author confirmation required before submission: state all funding sources and grant numbers, or "No funding was received".

Competing Interests

Author confirmation required before submission: disclose relevant financial and non-financial interests, or state that the authors have no competing interests to declare.

Ethics Approval

Not applicable. This study did not involve human participants, animals, or identifiable personal data.

Consent to Participate and Consent for Publication

Author confirmation required before submission: state "Not applicable" if no human participants or identifiable personal information were involved.

Data, Code, and Materials Availability

The HAMNet source code and supporting implementation materials are available through the project repository: https://github.com/nitin85011/HAMNET. The repository provides the model architecture, training and evaluation procedures, and the supporting files available at submission. However, the retained record does not include the exact run configuration, environment lockfile, system report, source revision, checkpoint checksum, or prediction archive needed to reproduce the reported numerical result exactly (Section 3.9). These checkpoint-linked materials must be deposited in the same repository or a permanent supplementary archive before submission.

The dataset and derived annotations used in this study are subject to the access conditions described in the repository. Where the original data cannot be redistributed, the repository provides the preprocessing and data-generation procedures needed to reproduce the derived inputs.

Trained model checkpoints, prediction outputs, machine-readable configuration files, environment specifications, and supplementary experimental results should be made available through the same repository or a permanent supplementary archive where permitted. Materials that cannot be redistributed publicly because of dataset, institutional, or third-party restrictions may be available from the corresponding author upon reasonable request.

Author Contributions

Author confirmation required before submission: provide a CRediT-style statement identifying each author's contributions.

References

[1] Alashban A, Alsadan A, Alhussainan NF, et al (2022) Single convolutional neural network with three layers model for crowd density estimation. IEEE Access 10:63823–63833

[2] Alhawsawi AN, Khan SD, Ur Rehman F (2024) Crowd counting in diverse environments using a deep routing mechanism informed by crowd density levels. Information 15(5):275

[3] Du Z, Shi M, Deng J, et al (2023) Redesigning multi-scale neural network for crowd counting. IEEE Transactions on Image Processing 32:3664–3678

[4] Guo M, Liu Z, Wen Z, et al (2025) Discrepancy-controlled region-adaptive learning: Handling intra-domain bias for crowd counting. IEEE Transactions on Circuits and Systems for Video Technology

[5] Jia X, Li N, Ling N, et al (2025) Stcc: Scale-aware transformer for crowd counting. Knowledge-Based Systems p 114992

[6] Khan A, Kadir K, Nasir H, et al (2022) Crowd counting and localization beyond density map. IEEE Access 10:133142–133151

[7] Kong W, Shen J, Li S, et al (2026) Haze-prior guided frequency-embedded attention learning for single-stage hazy-weather crowd counting. Information Processing & Management 63(5):104671

[8] Li N, Chen J, Jia X, et al (2026) Mslan: Multi-scale spatial location awareness network for crowd counting. Pattern Analysis and Applications 29:169. https://doi.org/10.1007/s10044-026-01748-2
[9] Li YC, Jia RS, Hu YX, et al (2024) A lightweight dense crowd density estimation network for efficient compression models. Expert Systems with Applications 238:122069
[10] Lin C, Hu X (2024) Efficient crowd density estimation with edge intelligence via structural reparameterization and knowledge transfer. Applied Soft Computing 154:111366
[11] Liu K, Dou Z, Wang F, et al (2025) Cross-level attention multi-scale context-enhanced crowd counting network for transportation cyber-physical systems. IEEE Transactions on Intelligent Transportation Systems
[12] Loshchilov I, Hutter F (2019) Decoupled weight decay regularization. In: International Conference on Learning Representations
[13] Phan DT, Gao J, Lu Y, et al (2024) Deo-net: Joint density estimation and object detection for crowd counting. IEEE Transactions on Instrumentation and Measurement 73:1–11
[14] Ptak B, Pieczyński D, Piechocki M, et al (2022) On-board crowd counting and density estimation using low altitude unmanned aerial vehicles—looking beyond beating the benchmark. Remote Sensing 14(10):2288
[15] Rohra A, Yin B, Kumar A, et al (2026) Multi-scale feature fusion with cross-view head re-identification module for crowd-counting. Pattern Analysis and Applications 29:112. https://doi.org/10.1007/s10044-026-01697-w
[16] Sharma S, Sharma D, Mundra A, et al (2026) Hybrid scale driven cascade cnn model for dense crowd counting and density estimation. Journal of Integrated Science and Technology 14(5):1570
[17] Sharma VK, Mir RN, Singh C (2023) Scale-aware cnn for crowd density estimation and crowd behavior analysis. Computers and Electrical Engineering 106:108569
[18] Shi W, Caballero J, Huszar F, et al (2016) Real-time single image and video super-resolution using an efficient sub-pixel convolutional neural network. In: Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition, pp 1874–1883
[19] Shorten C, Khoshgoftaar TM (2019) A survey on image data augmentation for deep learning. Journal of Big Data 6:60. https://doi.org/10.1186/s40537-019-0197-0
[20] Simard PY, Steinkraus D, Platt JC (2003) Best practices for convolutional neural networks applied to visual document analysis. In: Proceedings of the Seventh International Conference on Document Analysis and Recognition, pp 958–963, https://doi.org/10.1109/ICDAR.2003.1227801
[21] Simonyan K, Zisserman A (2015) Very deep convolutional networks for large-scale image recognition. arXiv preprint arXiv:14091556
[22] Song L, Li T, Cai Z, et al (2026) Fourier transform-based single domain generalization for crowd counting. Scientific Reports 16:11744

[23] Tong L, Li H, Kong W (2026) Gfaf-net: Global–local feature extraction and adaptive fusion network for rgb-t crowd counting in video surveillance systems. IEEE Transactions on Automation Science and Engineering 23:7869–7879
[24] Tripathy SK, Srivastava S, Bajaj D, et al (2024) A novel cascaded deep architecture with weak-supervision for video crowd counting and density estimation. Soft Computing 28(13):8319–8335
[25] Wang J, Fu H, Yang Y, et al (2026) Gru-enhanced multimodal crowd counting via modal-specific attention. Pattern Recognition p 113492
[26] Wang Z, Bovik AC, Sheikh HR, et al (2004) Image quality assessment: From error visibility to structural similarity. IEEE Transactions on Image Processing 13(4):600–612. https://doi.org/10.1109/TIP.2003.819861
[27] Woo S, Park J, Lee JY, et al (2018) CBAM: Convolutional block attention module. In: Proceedings of the European Conference on Computer Vision, pp 3–19
[28] Yang L, Guo Y, Sang J, et al (2022) A crowd counting method via density map and counting residual estimation. Multimedia Tools and Applications 81(30):43503–43512
[29] Yao B, Li Z, Yuan W, et al (2026) Lightweight and effective crowd counting and localization in diverse low-visibility conditions. Pattern Recognition p 113339
[30] Yin T, Wang J, Wang C, et al (2026) Multimodal individual counting: Robust crowd estimation under low-visibility conditions. Neurocomputing p 133289
[31] Yu F, Koltun V (2016) Multi-scale context aggregation by dilated convolutions. In: International Conference on Learning Representations
[32] Zhang Y, Zhou D, Chen S, et al (2016) Single-image crowd counting via multi-column convolutional neural network. In: Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition, pp 589–597
[33] Zhao Y, Pu L, Song H, et al (2026) Csafnet: Cross-modal spatial alignment and fusion network for rgb-t crowd counting. Pattern Recognition p 113250