MSP2P: Multi-Scale Point-based Approach for Optimal Crowd Localization Through Perspective Analysis

David Redó Nieto [1,2,*], Mikel Aramburu Retegui [1], Jorge García Castaño [1], and Antonio José Sánchez Salmerón [2]

[1] Fundación Vicomtech, Basque Research and Technology Alliance (BRTA), Donostia, Spain
[2] Instituto de Automática e Informática Industrial, Universitat Politècnica de València, Valencia, Spain
Email: dredo@vicomtech.org (D.R.N.); maramburu@vicomtech.org (M.A.R.); jgarcia@vicomtech.org (J.G.C.); ansanche@upv.edu.es (A.J.S.S)
*Corresponding author

***Abstract*—Image-based individual localization in densely populated scenes offers practical advantages beyond mere head counting, enabling a broader range of high-level tasks in crowd analysis. Crowd image data contain drastic changes in head sizes caused by the perspective effect. This specific challenge has not been addressed in the literature, as existing localization methods do not consider multi-scale features. To alleviate this issue, we propose a novel Multi-Scale Point-to-Point Network (MSP2P) in which a set of experts are in charge of predicting head locations a at different perspective levels. However, the training procedure requires ground-truth scale information for precise one-to-one matching. For this reason, we develop a simple yet effective method that uses neighbor density information to estimate the scale associated with each head location. Extensive experiments demonstrate that our method outperforms most state-of-the-art methods on relevant counting benchmarks without compromising performance.**

***Keywords*—crowd localization, multi-scale, crowd counting**

I. Introduction

Crowd monitoring is a sub-field of security and crowd surveillance that contains different tasks such as video monitoring, emergency management and crowd analysis [1]. The common denominator of these tasks is the use of crowd counting and crowd localization methods. The goal of crowd counting is to estimate the number of individuals in a crowded scene, while crowd localization aims to not only estimate the number of people but also to determine the position of each person's head within the scene. Crowd monitoring has several real-world applications across different domains. Safety in public events is one of the major areas where it is being used. It can help to avoid overcrowding, which can lead to dangerous situations like stampedes. In transportation services, these models are being used to monitor passenger numbers in real time at stations and airports. This data can be combined with tracking algorithms to provide crowd's flow to optimize routes and enhance security. For instance, analyzing human behavior in road health inspections systems [2].

Image-based crowd localization remains challenging due to factors such as low head resolution, which makes it difficult to distinguish individuals, especially in densely populated scenes; occlusions, where one person obstructs another; and high crowd density, where a large number of people stand close together or move in different directions within a confined space. While existing methods have been focused on mitigating these issues, the view perspective effect has not yet been addressed as shown in Fig. 1. This effect significantly influences crowd localization methods because they do not intrinsically account for the varying head sizes.

Fig. 1. Representation of the view perspective effect in high-density scenarios. The head size drastically changes from the bottom to the upper part of the image. Each color (red, green, and blue) represents the multiple scales required for optimal crowd localization.

In object detection tasks, the object scale is determined by the size of its bounding box, which is then used by object detection algorithms [3] to predict objects from different sizes at different levels. Extracting the scale or size of an object in this particular context is straightforward since this information is part of the ground

Manuscript received August 7, 2024; revised August 21, 2024; accepted September 4, 2024; published February 27, 2025.

truth data. Besides, in such task, the association process for each level in the network is quickly done by the Intersection over Union (IoU). In contrast, popular crowd localization datasets [4–6] only provide a point representing each person's head in the crowd, which lacks scale information.This lack of scale information is a crucial issue since there is no prior knowledge regarding the evaluated scene, pushing existing methods to be as accurate as possible in all kinds of scenarios.

Crowd localization has been addressed from different perspectives, such as detection methods [7, 8] in which they try to predict the bounding box of each person's head. Compared to current approaches, these methods do not show satisfactory results since predicting bounding boxes in highly dense scenarios can lead to inaccurate results due to the heuristic post-processing step that removes negative predictions. Density-map methods quickly gained popularity in crowd counting since they have reported better results than the previous approaches.

Typically, these methods do not provide individual locations since they focus on predicting a more accurate density map [9–11]. However, some methods attempt to estimate the point localization from the density map with different approaches, such as creating a topological constraint during the training process [12] or using a connected components algorithm [13] to extract blobs from the scene.

For this reason, a new trend has been shown recently for methods that predict the coordinates of each individual in the crowd since it is more suitable for higher-level tasks. Within this new trend, Dingkang Liang *et al.* [14] tried to solve the problem with transformers, similar to Detection Transformer (DETR) [15], as it was a breakthrough in object detection tasks. Nevertheless, it still suffers from two problems: the wide range of scales that can appear in the scene and the higher inference time compared to the Convolutional Neural Networks (CNNs) counterpart. Another solution for predicting the location of each individual is the Point-to-Point Network (P2PNet) [16], which is based on a set of point proposals and a one-to-one association between the ground truth and the prediction. Similarly to the previous method, the whole network fails to predict head coordinates at a wide range of scales, caused by strong perspective as the network only uses one level for point regression.

To address this issue, we propose a Multi-Scale Point-to-Point Network (MSP2P). Specifically, each level of the network contains an expert that learns and predicts points at different scales. Moreover, we develop a scale estimation method that leverages ground truth points and their neighbors' distances to estimate the scale associated with each individual, which will be included in the target association process for one-to-one matching.

In summary, the main contributions of this paper are:

- We present MSP2P, a multi-scale architecture for a point-to-point framework, leveraging the Feature Pyramid Network (FPN) usage by combining the low-level spatial features and the high-level semantic features, including different prediction heads for the different scales.
- We develop a method for estimating the optimal scale parameter of each head in the scene based on its neighbors' distances, which is utilized in one-to-one matching.
- Numerous experiments are carried out to validate that the method achieves state-of-the-art results, making the model robust to dense scenes and high variations in scale, including cross-domain validation in order to manifest the method's ability to generalize between datasets.

II. Related Works

A. Counting by Density Map Methods

Currently, most state-of-the-art methods are density-map-based in which the final count is directly estimated from the predicted map [17]. These methods suffer from extreme overlap in dense regions, so current methods put effort into alleviating this effect with different approaches. Designing better density maps is crucial for better estimation, for example, Xu *et al.* [9] proposes to automatically scale dense regions to reduce the number of overlapped blobs. Some methods propose the creation of a new target map for learning in which [10] designs a new map based on the Focal Inverse Distance Transform, whereas Liu *et al.* [18] introduces the Local Counting Map.

Others leverage multi-scale architecture [19] to merge the estimation from different levels, while Hu *et al.* [11] uses Neural Architecture Search (NAS) to discover the multi-scale design of the counting model automatically. On the other hand, Tran *et al.* [20] employs Vision Transformer (ViT) [21] in non-overlapping patches of the image to estimate the number of people in each cell. Following the idea of replacing traditional convolutional layers, Cheng *et al.* [22] proposes to substitute the convolutional operation with Gaussian convolution to mimic the style throughout the whole learning process instead of merely generating it in the final step. However, even though these methods achieve competitive results or, in some cases, better results, they do not provide localization of individuals, which is essential for many other tasks.

B. Counting by Localization Methods

This group is formed by methods that perform counting by first providing the localization of individuals. High accuracy object detectors such as Faster-RCNN [23], inspired the development of detection models [7, 8] for solving the counting problem. Nevertheless, since only point-level annotations are available for most datasets, these methods rely on estimating bounding boxes for people's heads, which lead to inaccurate results in high-density scenarios with large variations of scales. Other methods rely on a post-processing step after generating a density-map [24] with no remarkable increase in accuracy. Recently, inspired by the use of transformers in object detection, Liang *et al.* [14] proposes to adapt DETR [15] to crowd localization. However, it does not detect heads with large scales. On the other hand, P2PNet [15] directly predicts a set of points proposals with a purely point-based

framework. Since it only uses one network level for prediction, it still suffers from undetected large-scale head issues.

III. Method

Fig. 2 details the pipeline proposed for the Multi-Scale Point-to-Point Network (MS2P2). We start presenting the architecture for multi-scale prediction and the associated framework in Section III.A. Then, in Section III.B, we present in-depth the developed method for estimating the scale parameter. In Section III.C, it is explained how the one-to-one association strategy between the different levels in the network works with the novel scale parameter. Lastly, the loss function is presented in Section III.D.

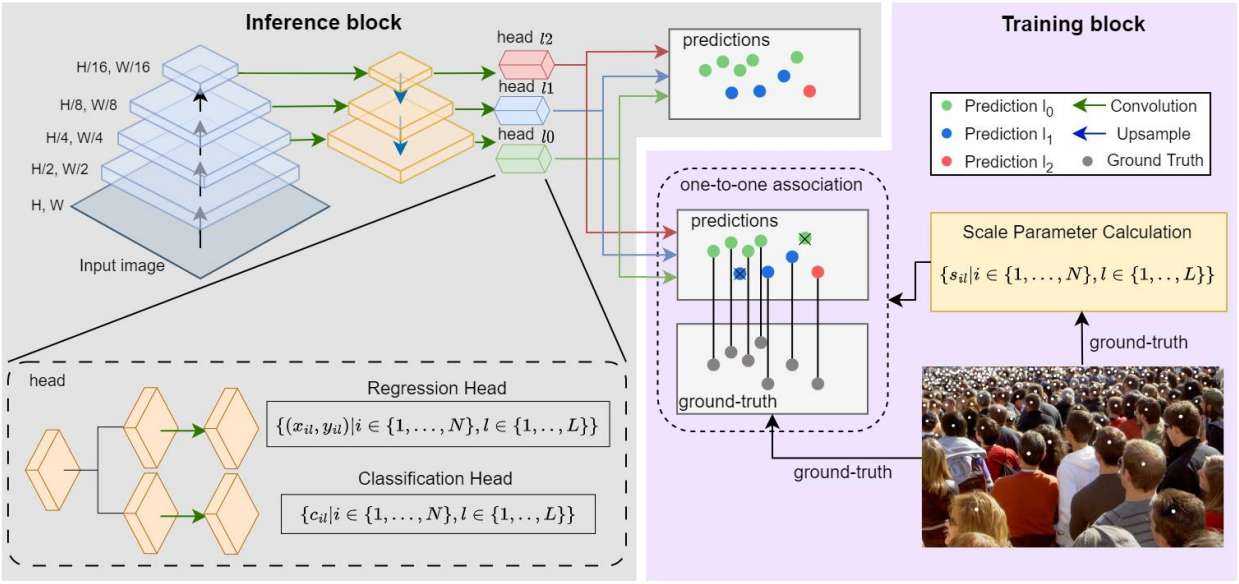


Fig. 2. Overview of the MSP2P architecture for training and inference. The network is built upon a Feature Pyramid Network with independent prediction heads at each level in order to predict heads at different scales. Each head, representing an expert for a specific scale, outputs two sets of predictions: (1) a set of proposal points and (2) their confidence score. Note that the scale parameter calculation is an offline pre-training step whose information is only used during training in the one-to-one association.

A. The MSP2Pet Model

Following the strategy suggested in object detectors [3, 25], in which objects with different scales are predicted, MSP2P architecture is composed of a CNN backbone, which serves as the foundational feature extractor. The backbone is followed by a Feature Pyramid Network (FPN) [3], a structure that is specifically designed to preserve and enhance the spatial hierarchies inherent in the image data. The FPN enables the model to analyze the image at multiple scales simultaneously, which is critical for accurately detecting heads that vary greatly in size due to perspective distortions and crowd density. Then, a regression head is assigned to each pyramid level in order to predict people's heads at different scales. Thus, these regression heads act as specialized experts, each trained to predict head locations at a particular scale.

Formally, we use $l \in \{1,\dots,L\}$ to represent the prediction head at level l of the pyramid with L levels. These levels are variable depending on the scale range of the different individuals in the dataset. Following the notation introduced by Song *et al.* [16], for any given image with N individuals, $P = \{ p_i \mid i \in \{1,\dots,N\} \}$ represents the collection of ground truth points that indicates the center point $p_i = (x_i, y_i)$ of the i-th individual's head. For each prediction head l in the network, the trained model outputs two subsets of predictions: $\hat{P}_l = \{ \widehat{p_{jl}} \mid j \in \{1,\dots,M_l\}, l \in \{1,\dots,L\}\}$ and $\{\hat{C}_l = \{ \widehat{c_{jl}} \mid j \in \{1,\dots,M_l\}, l \in \{1,\dots,L\}\}$, where $\widehat{c_{jl}}$ represents the confidence score of the predicted point $\widehat{p_{jl}}$ at level l in which the number of predicted individuals is M_l. Then, previous subsets can be grouped into $\hat{P} = \{\hat{P}_l \mid l \in L\}$, $\hat{C} = \{\hat{C}_l \mid l \in L\}$ and $\hat{M} = \{\hat{M}_l \mid l \in L\}$. Unlike regression-based methods, where all points are predicted at one level, our network predicts points at different levels. So our goal is to match a predicted point $\widehat{p_{jl}}$ to its ground truth p_i not only using the distance between them and its confidence score $\widehat{c_{jl}}$ but also including scale information. For this reason, we introduce the set $S = \{s_{il} \mid i \in \{1,\dots,N\}, l \in \{1,\dots,L\}\}$ that represents the scale parameter associated to each p_i in Section III.B. This strategy allows the network to exploit the hierarchical structure of CNNs in multi-level networks to learn multi-scale feature representations.

B. Scale Parameter Calculation

Three steps are required for calculating the scale parameter used in training. Note that this is an offline method performed before training since it only uses the ground truth as input. This parameter will be used during the one-to-one matching strategy in Section III.C.

1) Neigbours dispersion step

First, we introduce a dispersion estimation d_i for each ground truth point p_i. An Area of Interest (AoI) of radius R is placed at each point to find its neighbors as shown in

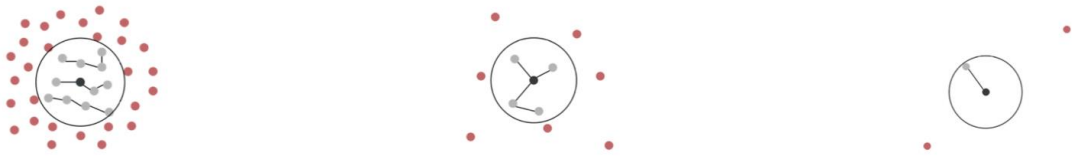
Fig. 3, in which the set $G = g_k$ represents the neighbors inside the AoI. Then, the dispersion d_i for p_i is calculated as:

$$d_i = \begin{cases} \frac{1}{|G|}\sum_{g_k \in G} min(g_k)/R, & if\ G \neq \emptyset, \\ 1, & otherwise, \end{cases} \quad (1)$$

where *min(gk)* is calculated as:

$$\min(g_k) = \min\{d(g_k, g_t) \mid g_t \in G, t \neq k\} \quad (2)$$

in which $d(g_k, g_t) = ||g_k - g_t||_2$ denotes the Euclidean distance.



(a) High density area (b) Medium density area (c) Low density area

Fig. 3. Density scenarios representing different dispersion values (considering L = 3). (a) A high-density area will have a low di value since the distance between neighbors is close. (b) An intermediate density area defines the scales that can not be considered small or large. (c) Typically, sparse points are located in areas with high-scale heads, producing a high di value.

2) Correction step

Commonly, the training sets contain highly crowded scenes in which the density of an area is directly related to the size of the people in it, meaning that areas with people far from the camera will appear closer together, therefore smaller than those close to the camera. However, there are scenarios where this hypothesis is not met since some images are not fully covered by people. For this reason, we consider a correction step for those cases in which an empty space does not reflect the real distribution.

When the neighbor dispersion d_i is above the threshold θ, it could mean one of these two options: (i) a person is far from the point of view with no neighbor around, (ii) or a person is close to the point of view. To solve this ambiguity and using the ground truth as a starting point, we utilize Segment Anything [26] to obtain the area considered as head inside the AoI to provide extra information about the size.

Assuming that the segmented head inside the AoI is *S*, the refined dispersion d_i' for each point is calculated as follows:

$$d_i' = \begin{cases} d_i * \frac{S}{AoI}, & if\ d_i > \theta \\ d_i & otherwise \end{cases} \quad (3)$$

3) Scale parameter step

Each level should respond differently to the same input since each level is assigned to a single scale [25]. Next, once it is obtained the refined dispersion from the previous step, it is possible to calculate the scale parameter s_{il} used during the matching step in Section III.C as:

$$s_{il} = e^{\left(-\frac{\left(d_i' - \frac{l}{L-1}\right)^2}{p}\right)} \quad (4)$$

where variable *p* is calculated as:

$$p = -\frac{1}{(L-1)^2 \cdot ln(0.2)} \quad (5)$$

Fig. 4 is a graphical example of how these steps work. In this example, we assume that L = 3, and we assign a color to each level l. As explained in Section III.C, during the matching process, this parameter is used along with the distance and the confidence; however, for the purpose of this example, we assign the level l to p_i with the highest response in Eq. (4).

C. Proposal Matching

In order to train the model, we need to match the ground truth P to the predictions $\hat{P}$ using a one-to-one matching strategy. Assuming that N is the number of people in an image and M is the total number of predicted individuals, the cost matrix D will be of shape NxM. Since each level has its own predictions $\hat{P}_l$, the cost matrix for level l is calculated as:

Fig. 4. Graphical example of how the scale estimation method works (considering L = 3). Additionally, for visualization purposes, each ground truth p_i is assigned to the level with the highest response from Eq. (4). Top row shows the response without the correction step. In contrast, bottom row shows the response with the correction step.

$$\mathcal{D}_\ell(\mathcal{P},\hat{\mathcal{P}}) = \left(\tau\,||p_i - \widehat{p_{jl}}|\,|_2 - \widehat{c_{jl}} - \gamma s_{il}\right)_{i\in N, j\in M_l, l\in L} \quad (6)$$

where $\|\cdot\|_2$ represents the l2 distance, $\widehat{p_{jl}}$ is the predicted point along with its confidence $\widehat{c_{jl}}$ and s_{il} denotes the scale parameter for p_i at level l calculated in Eq. (4). τ and γ are weight terms to balance the influence of the pixel distance and the scale estimation, respectively.

Since only one p_i can be assigned to one $\widehat{p_{jl}}$, these matrices are concatenated to create the cost matrix $D(P,\hat{P})$, which is used by the Hungarian algorithm [27] as a matching strategy. This matching strategy allows the network to learn the optimal scale level for each head in the image. Note that this matcher is only used during training.

D. Loss Function

Once the matching step has linked each ground truth to a target, we calculate the loss for point regression and point classification. Regarding point regression L_{loc}, we employ the common MSE loss:

$$L_{loc} = \left|\left|p_i - \widehat{p_{jl}}\right|\right|_2 \quad (7)$$

where p_i is the i-th ground truth matched with the prediction $\widehat{p_{jl}}$ from level l. We utilize the focal loss [24] as the classification loss L_{cls}, which defines the total loss as:

$$L = L_{loc} + \lambda L_{cls} \quad (8)$$

where the weight term λ balances the effect of regression loss.

IV. Experimental Results

A. Datasets

Extensive experiments have been conducted on our method against four well-known, publicly available datasets, which are described below:

ShanghaiTech [4]. It is composed of two independent subsets: PartA and PartB. PartA contains crowded images from different perspectives collected from the Internet, while PartB contains images with different densities of people in a busy street, similar to a surveillance camera. PartA consists of 300 images for training and 182 for testing, whereas PartB contains 400 images for training and 316 for testing.

UCF_CC_50 [6]. It is a small but challenging dataset, with only 50 images collected from the Internet, as it contains large variations of people. Following [6], a five-fold cross-validation has been implemented for evaluation.

UCF_QNRF [5]. It is a dense dataset containing over 1.2 million annotated instances in 1535 images, from which 1201 images are for training and 334 for testing. Apart from the high density, it is also challenging since it contains diverse viewpoints and lighting variations.

JHU++ [28]. It is a highly crowded dataset in which the total count of people in each image varies from 0 to 25791. It comprises three subsets: the training set with 2272 images, the validation set with 500 images, and lastly, the testing set with 1600 images.

B. Implementation Details

We augment the datasets for training using different strategies such as random scaling, random cropping, and horizontal flipping as implemented in [14, 16]. Random cropping is performed after random scaling with a crop size of 256×256 for ShanghaiTech and UFC_CC_50 datasets, while the crop size for UCF_QNRF and JHU++ is 512×512. The horizontal flipping augmentation is performed with a probability of 0.5. Since the UCF QNRF and JHU++ datasets contain extremely large images, we set the maximum size to 1920, keeping the original aspect ratio.

The batch size used for training is 8. The radius R for the Area of Interest (AoI) is set to 45 for ShanghaiTech and 65 for UCF QNRF, UFC CC 50 and JHU++. The confidence threshold is set to 0.5 to filter the "non-head" class. The weight terms τ and γ in the cost matrix are set to 5e-2 and 2, respectively. The parameter θ is set to 0.5. We used Adam [29] with 1e-4 as the learning rate to optimize the model parameters.

C. Evaluation Metrics

Counting Metrics. For this task, the Mean Absolute Error (MAE) and the Mean Square Error (MSE) are used, and they are defined as:

$$MAE = \frac{1}{N}\sum_{i=1}^{N}|P_i - G_i| \quad (9)$$

$$MSE = \sqrt{(1/N\sum_{i=1}^{N}|P_i - G_i|^2)} \quad (10)$$

where N is the total number of images, P_i and G_i correspond to the predicted count and the ground truth of the *i-th* image, respectively.

Localization Metrics. Precision, Recall, and F1-Measure are the localization metrics used in this work, following [4, 5]. To obtain the True Positives (TP), the distance between ground truth point P and a predicted point $\hat{P}$ must be less than a predefined threshold σ.

The ShanghaiTech dataset is evaluated using two thresholds, σ = 4 and σ = 8. Regarding the UCF QNRF dataset, we use a range of thresholds from [1, 100] as established in [5].

D. Comparison with State-of-the-Art

1) Crowd Counting

In this section, we analyze and compare the performance of our method in the counting task with several methods, including those that do not output localization information as shown in Table I. Density-map methods have been known for outperforming localization-based methods. However, it has been demonstrated that MSP2P yields comparable results to state-of-the-art approaches while providing individuals' locations.

TABLE I. COUNTING ACCURACY OF OUR METHOD ON FIVE CHALLENGING DATASETS COMPARED WITH STATE-OF-THE-ART METHODS

Method	Localization	Shanghai A		Shanghai B		UCF-QNRF		UCF-CC_50		JHU++	
		MAE	MSE	MAE	MSE	MAE	MSE	MAE	MSE	MAE	MSE
AMSNet [11]	No	56.7	93.4	6.7	10.2	101.8	163.2	208.6	296.3	-	-
AMRNet [18]	No	61.6	98.4	7.0	11.0	86.6	152.2	184.0	265.8	-	-
NoisyCC [30]	No	61.9	99.6	7.4	11.8	85.8	150.6	-	-	-	-
LoViTCrowd [20]	No	54.8	**80.9**	8.6	13.8	87.0	**141.9**	-	-	-	-
S-DCNet [31]	No	59.8	100.0	6.8	11.5	84.8	142.3	-	-	62.1	268.9
Gaunet [22]	No	54.8	89.1	6.2	9.9	**81.6**	153.7	186.3	256.5	58.2	245.1
HMoDE [19]	No	**54.4**	87.4	**6.2**	**9.8**	-	-	**159.6**	**241.6**	**55.7**	**214.6**
LSC-CNN [7]	Yes	66.4	117.0	8.1	12.7	120.5	218.2	225.6	302.7	-	-
GL [32]	Yes	61.3	95.4	7.3	11.7	84.3	147.5	-	-	59.9	259.5
TopoCount [12]	Yes	61.2	104.6	7.8	13.7	89.0	159.0	184.1	233.1	60.9	267.4
AutoScale_loc [9]	Yes	65.8	112.1	8.6	13.9	104.4	174.2	-	-	85.6	356.1
CLTR [14]	Yes	56.9	95.2	6.5	10.6	85.8	141.3	-	-	59.5	**240.6**
P2P* [16]	Yes	53.6	94.1	6.3	10.4	93.3	164.3	182.8	263.9	60.8	250.2
MSP2P (ours)	Yes	**53.2**	**87.9**	**6.1**	**9.9**	**84.1**	**140.5**	**158.8**	**239.2**	**57.2**	245.0

Note: * represents that the network was trained by ourselves with a fixed number of epochs. Bold represents the best results for each family of methods.

Compared to P2P [16], it is also shown in the same table that our method MSP2P is able to reduce by a significant margin both counting metrics on UCF-QNRF [5], UCF_CC_50 [6] and JHU++ [28] datasets, which are the ones with more complicated scenes containing a significant variation of crowd numbers with large a scale of people. Compared to the other methods, MSP2P competes with the transformer-based methods [14, 20]. Regarding the ShanghaiTech dataset [4], our method achieves similar results to P2P, improving the MSE metric in Part A. Moreover, even though that Part B contains more sparse scenes with fewer scale variations, our method still achieves competitive results. Regarding the transformer-based models, in Part A MSP2P obtains a 5% reduction compared to CLTR [14], which is the next best method that provides position information.

2) Crowd localization

First, crowd localization evaluation is shown in Tables II and III in which it is compared with other state-of-the-art methods [7, 9, 12, 14]. When using a low threshold (σ = 4) in ShanghaiTech Part A, our MSP2P improves by 3% the F1-measure of P2P [16] and AutoScale [9] and outperforms the transformer-based CLTR [14] at least 15%.

When the threshold is less strict (σ = 8), the results are more balanced, and it improves 2% the F1-Measure on the test set. For the high-dense dataset (Table III), UCF QNRF (σ = [1, 100]), our method achieves the best Avg. Precision and F1-Measure. These results show evidence that providing scale information during training helps the network to improve the localization of individuals in the images. Fig. 5 illustrates some failure cases in localization from the baseline (top row) in which large-scale heads are not detected and, in some cases, estimates multiple points for the same individual.

TABLE II. LOCALIZATION RESULTS FOR SHANGHAITECH PART A [3]. IT IS EVALUATED WITH TWO THRESHOLDS: Σ = 4 AND Σ = 8

Method	σ = 4			σ = 8		
	Prec.	Rec	F1	Prec.	Rec.	F1
LSC-CNN [7]	33.4	31.9	32.6	63.9	61.0	62.4
TopoCount [12]	41.7	40.6	41.1	74.6	72.7	73.6
AutoScale [9]	56.2	54.2	55.2	74.4	71.7	73.0
CLTR [14]	43.6	42.7	43.2	74.9	73.5	74.2
P2P* [16]	56.2	54.9	55.6	77.1	76.2	76.6
MSP2P	**58.7**	**58.3**	**58.5**	**79.6**	**77.7**	**78.6**

Note: * represents that the network was trained by ourselves with a fixed number of epochs.

TABLE III. LOCALIZATION RESULTS FOR UCF-QNRF [4]. IT IS EVALUATED USING A RANGE OF THRESHOLDS FROM [1, 100]

Method	Precision	Recall	F1-Measure
LSC-CNN [7]	75.84	74.69	75.26
TopoCount [12]	81.77	78.96	80.34
AutoScale [9]	81.31	75.75	78.43
CLTR [14]	82.22	**79.75**	80.97
P2P* [16]	80.91	78.01	79.43
MSP2P (ours)	**82.51**	79.56	**81.01**

Note: * represents that the network was trained by ourselves with a fixed number of epochs.

Fig. 5. Comparison between P2P [15] (top row) and our MSP2P (bottom row) when localizing large-scale heads. As seen in the top row inside the yellow regions, P2P fails at localizing people and, in some cases, predicts multiple points for the same individual, as seen in the third column.

E. Ablation Study

In this section, we study the effect of different parameters on the ShanghaiTech Part A and UCF-QNRF datasets.

Effect of the number of prediction heads. The number of levels L in a network specifies the different scales in which the network is able to predict. As explained in Sec. III.B, the number of levels can affect the scale parameter used during training. This effect can be seen in Table IV, where the best results are obtained when L = 3.

TABLE IV. EFFECT OVER THE COUNTING RESULTS DEPENDING ON THE NUMBER OF PREDICTION LAYERS

Levels (L)	SHTech Part A		UCF-QNRF	
	MAE	MSE	MAE	MSE
2	53.4	90.5	88.8	151.8
3	**53.2**	**87.9**	**84.1**	**143.5**
4	60.6	90.4	100.4	173.7

Effect of AoI size. The radius R determines the Area of Interest, regulating the number of neighbors inside the AoI. In Table V, we study the effect of its size on the counting results. We test three different sizes in each dataset, and the best results are obtained when the AoI is similar in size to large-scale heads in that particular dataset.

Effect of correction step. In Table VI, we show how the correction step has a significant effect on the count estimation. Correcting ambiguous points creates a smoother transition between scales, allowing the network to learn better representations for different scales.

TABLE V. EFFECT OVER THE COUNTING RESULTS DEPENDING ON THE SIZE OF THE AREA OF INTEREST

Radius (R)	SHTech Part A		UCF-QNRF	
	MAE	MSE	MAE	MSE
2	53.4	90.5	88.8	151.8
3	**53.2**	**87.9**	**84.1**	**143.5**
4	60.6	90.4	100.4	173.7

TABLE VI. EFFECT OF INCORPORATING THE CORRECTION STEP

Correction step	SHTech Part A		UCF-QNRF	
	MAE	MSE	MAE	MSE
False	56.3	91.2	96.9	168.7
True	**53.2**	**87.9**	**84.1**	**143.5**

V. CONCLUSION

In this work, we have proposed a multi-scale architecture point-to-point network (MSP2P) specifically designed to address the challenge of predicting accurate head counts in densely populated scenarios with high perspective variations. This network includes a set of experts where each one has been specifically trained to estimate head locations in a range of head sizes. Additionally, we have presented a novel method for determining the optimal scale parameter for each annotated head, which aids the neural network in selecting the most suitable level for each ground truth point, specifically during the one-to-one association step in the training phase. This simple yet highly effective approach outperforms state-of-the-art results, improving not only counting and localization accuracy but also exhibiting superior generalization capabilities across diverse datasets. Visual results demonstrate that the proposed method mitigates the view perspective effect.

CONFLICT OF INTEREST

The authors declare no conflict of interest.

AUTHOR CONTRIBUTIONS

D.R.N. conducted the research and wrote a draft version of the study; M.A.R prepared the experiments and analyzed the results; J.G.C and A.J.S.S finished the paper writing and revision; all authors had approved the final version.

FUNDING

The work described in this paper is performed in the H2020 project STARLIGHT ("sustainable Autonomy and Resilience for LEAs using AI against High Priority Threats"). This project has received funding from the European Union's Horizon 2020 research and innovation program under grant agreement No 101021797.

REFERENCES

[1] X. Wang and C.-C. Loy, "Deep learning for scene-independent crowd analysis," in *Group and Crowd Behavior for Computer Vision*, Academic Press, 2017, ch. 10, pp. 209–252.

[2] T. Siriborvornratanakul, "Human behavior in image-based Road Health Inspection Systems despite the emerging AutoML," *Journal of Big Data*, pp. 2196–1115, 2022.

[3] T. Lin, P. Dollár, R. B. Girshick, K. He, B. Hariharan, and S. J. Belongie, "Feature pyramid networks for object detection," in *Proc. IEEE Conference on Computer Vision and Pattern Recognition*, IEEE, 2017, pp. 936–944.

[4] Y. Zhang, D. Zhou, S. Chen, S. Gao, and Y. Ma, "Single-image crowd counting via multi-column convolutional neural network," in *Proc. 2016 IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*, IEEE Computer Society, 2016, pp. 589–597.

[5] H. Idrees, M. Tayyab, K. Athrey, D. Zhang, S. Al-Máadeed, N. M. Rajpoot, and M. Shah, "Composition loss for counting, density map estimation and localization in dense crowds," in *Proc. Computer Vision—ECCV 2018*, Lecture Notes in Computer Science, Springer, 2018, 11206, 544–559.

[6] H. Idrees, I. Saleemi, C. Seibert, and M. Shah, "Multi-source multi-scale counting in extremely dense crowd images," in *Proc. 2013 IEEE Conference on Computer Vision and Pattern Recognition*, 2013, pp. 2547–2554.

[7] D. B. Sam, S. V. Peri, M. N. Sundararaman, A. Kamath, and R. V. Babu, "Locate, size, and count: Accurately resolving people in dense crowds via detection," *IEEE Trans. Pattern Anal. Mach. Intell.*, vol. 43, no. 8, pp. 2739–2751, 2021.

[8] Y. Liu, M. Shi, Q. Zhao, and X. Wang, "Point in, box out: Beyond counting persons in crowds," in *Proc. IEEE Conference on Computer Vision and Pattern Recognition*, Computer Vision Foundation / IEEE, 2019, pp. 6469–6478.

[9] C. Xu, D. Liang, Y. Xu, S. Bai, W. Zhan, X. Bai, and M. Tomizuka, "Autoscale: Learning to scale for crowd counting," *International Journal of Computer Vision*, vol. 130, no. 2, pp. 405–434, 2022.

[10] D. Liang, W. Xu, Y. Zhu, and Y. Zhou, "Focal inverse distance transform maps for crowd localization," *IEEE Transactions on Multimedia*, pp. 1–13, 2022.

[11] Y. Hu, X. Jiang, X. Liu, B. Zhang, J. Han, X. Cao, and D. S. Doermann, "Nas-count: Counting-by-density with neural architecture search," in *Proc. Computer Vision—ECCV 2020*, Lecture Notes in Computer Science, Springer, 2020, 12367, pp. 747–766.

[12] S. Abousamra, M. Hoai, D. Samaras, and C. Chen, "Localization in the crowd with topological constraints," in *Proc. Thirty-Fifth AAAI Conference on Artificial Intelligence*, AAAI Press, 2021, pp. 872–881.

[13] M. Zand, H. Damirchi, A. Farley, M. Molahasani, M. A.Greenspan, and A. Etemad, "Multiscale crowd counting and localization by multitask point supervision," in *Proc. 2022 IEEE International Conference on Acoustics, Speech and Signal Processing (ICASSP)*, 2022, pp. 1820–1824.

[14] D. Liang, W. Xu, and X. Bai, "An end-to-end transformer model for crowd localization," in *Proc. Computer Vision—ECCV 2022*, Lecture Notes in Computer Science, Springer, 2022, 13661, pp. 38–54.

[15] N. Carion, F. Massa, G. Synnaeve, N. Usunier, A. Kirillov, and S. Zagoruyko, "End-to-end object detection with transformers," in *Proc. Computer Vision—ECCV 2020*, Lecture Notes in Computer Science, Springer, 2020, 12346, pp. 213–229.

[16] Q. Song, C. Wang, Z. Jiang, Y. Wang, Y. Tai, C. Wang, J. Li, F. Huang, and Y. Wu, "Rethinking counting and localization in crowds: A purely point-based framework," in *Proc. the IEEE/CVF International Conference on Computer Vision (ICCV)*, October 2021, pp. 3365–3374.

[17] V. S. Lempitsky and A. Zisserman, "Learning to count objects in images," in *Proc. the 23rd International Conference on Neural Information Processing Systems*, Curran Associates, Inc., 2010, vol. 1, pp. 1324–1332.

[18] X. Liu, J. Yang, W. Ding, T. Wang, Z. Wang, and J. Xiong, "Adaptive mixture regression network with local counting map for crowd counting," in *Proc. Computer Vision—ECCV 2020*, Lecture Notes in Computer Science, Springer, 2020, 12369, pp. 241–257.

[19] Z. Du, M. Shi, J. Deng, and S. Zafeiriou, "Redesigning multi-scale neural network for crowd counting," *IEEE Transactions on Image Processing*, 2023.

[20] N. H. Tran, T. D. Huy, S. T. M. Duong, N. Phan, D. H. Hung, C. D. T. Nguyen, T. H. Bui, and S. Q. H. Truong, "Improving local features with relevant spatial information by vision transformer for crowd counting," in *Proc. 33rd British Machine Vision Conference*, BMVA Press, 2022, 729.

[21] A. Dosovitskiy, L. Beyer, A. Kolesnikov, D. Weissenborn, X. Zhai, T. Unterthiner, M. Dehghani, M. Minderer, G. Heigold, S. Gelly, J. Uszkoreit, and N. Houlsby, "An image is worth 16×16 words: Transformers for image recognition at scale," in *Proc. 9th International Conference on Learning Representations*, OpenReview.net, 2021.

[22] Z.-Q. Cheng, Q. Dai, H. Li, J. Song, X. Wu, and A. G. Hauptmann, "Rethinking spatial invariance of convolutional networks for object counting," in *Proc. the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, pp. 19638–19648, 2022.

[23] S. Ren, K. He, R. Girshick, and J. Sun, "Faster R-CNN: Towards real-time object detection with region proposal networks," in *Proc. Advances in Neural Information Processing Systems*, Curran Associates, Inc., 2015, 28.

[24] I. H. Laradji, N. Rostamzadeh, P. O. Pinheiro, D. Vazquez, and M. Schmidt, "Where are the blobs: Counting by localization with point supervision," in *Proc. the European Conference on Computer Vision (ECCV)*, 2018, pp. 547–562.

[25] T.-Y. Lin, P. Goyal, R. Girshick, K. He, and P. Doll´ar, "Focal loss for dense object detection," in *Proc. the IEEE International Conference on Computer Vision*, 2017, pp. 2980–2988.

[26] A. Kirillov, E. Mintun, N. Ravi, H. Mao, C. Rolland, L. Gustafson, T. Xiao, S. Whitehead, A. C. Berg, W.-Y. Lo, P. Doll´ar, and R. Girshick, "Segment anything," arXiv preprint, arXiv:2304.02643, 2023.

[27] H. W. Kuhn, "The hungarian method for the assignment problem," *Naval Research Logistics (NRL)*, vol. 52, no. 1, pp. 7–21, 2005.

[28] V. A. Sindagi, R. Yasarla, and V. M. Patel, "Jhu-crowd++: Large-scale crowd counting dataset and a benchmark method," *IEEE Transactions on Pattern Analysis and Machine Intelligence*, vol. 44, no. 5, pp. 2594–2609, 2020.

[29] D. P. Kingma and J. Ba, "Adam: A method for stochastic optimization," in *Proc. 3rd International Conference on Learning Representations*, 2015.

[30] J. Wan and A. B. Chan, "Modeling noisy annotations for crowd counting," in *Proc. Advances in Neural Information Processing Systems*, 2020.

[31] H. Xiong and A. Yao, "Discrete-constrained regression for local counting models," in *Proc. European Conference on Computer Vision*, Springer, 2022, pp. 621–636.

[32] J. Wan, Z. Liu, and A. B. Chan, "A generalized loss function for crowd counting and localization," in *Proc. the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, 2021, 1974–1983.

Copyright © 2025 by the authors. This is an open access article distributed under the Creative Commons Attribution License (CC-BY-4.0), which permits use, distribution and reproduction in any medium, provided that the article is properly cited, the use is non-commercial and no modifications or adaptations are made.