NAS-Count: Counting-by-Density with Neural Architecture Search

Yutao Hu[1] *, Xiaolong Jiang[4] *, Xuhui Liu[1], Baochang Zhang[5],
Jungong Han[6], Xianbin Cao[1,2,3] †, and David Doermann[7]

[1] School of Electronic and Information Engineering,
Beihang University, Beijing, China
[2] Key Laboratory of Advanced Technologies for Near Space Information Systems,
Ministry of Industry and Information Technology of China
[3] Beijing Advanced Innovation Center for Big Data-Based Precision Medicine,China
[4] YouKu Cognitive and Intelligent Lab, Alibaba Group
[5] Beihang University, Beijing, China
[6] Computer Science Department, Aberystwyth University, SY23 3FL, UK
[7] Department of Computer Science and Engineering,
University at Buffalo, New York, USA
{huyutao, bczhang, xbcao}@buaa.edu.cn, xainglu.jxl@alibaba-inc.com,
xuhui_cc@126.com, jungonghan77@gmail.com, doermann@buffalo.edu

Abstract. Most of the recent advances in crowd counting have evolved from hand-designed density estimation networks, where multi-scale features are leveraged to address the scale variation problem, but at the expense of demanding design efforts. In this work, we automate the design of counting models with Neural Architecture Search (NAS) and introduce an end-to-end searched encoder-decoder architecture, Automatic Multi-Scale Network (AMSNet). Specifically, we utilize a counting-specific two-level search space. The encoder and decoder in AMSNet are composed of different cells discovered from micro-level search, while the multi-path architecture is explored through macro-level search. To solve the pixel-level isolation issue in MSE loss, AMSNet is optimized with an auto-searched Scale Pyramid Pooling Loss (SPPLoss) that supervises the multi-scale structural information. Extensive experiments on four datasets show AMSNet produces state-of-the-art results that outperform hand-designed models, fully demonstrating the efficacy of NAS-Count.

Keywords: Crowd Counting, Neural Architecture Search, Multi-scale

1 Introduction

Crowd counting, aiming to predict the number of individuals in a scene, has wide applications in the real world and receives considerable attention [63, 56, 57]. With advanced occlusion robustness and counting efficiency, counting-by-density

*Contribute equally

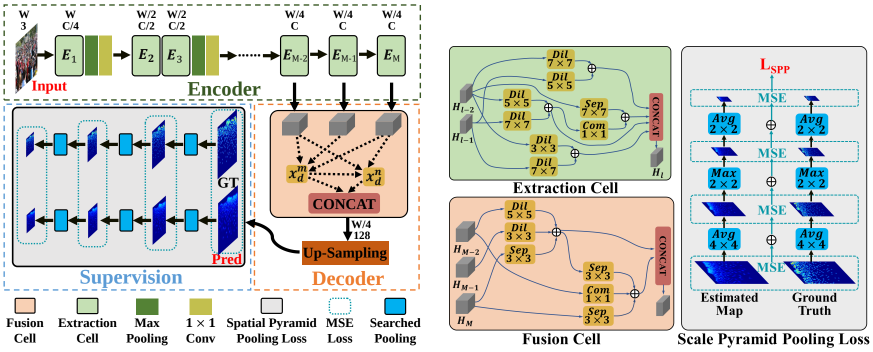
†Corresponding author

[34, 82, 10, 29] has become the method-of-choice over others related techniques [35, 21, 12, 28, 12]. These techniques estimate a pixel-level density map and count the crowd by summing over pixels in the given area.

Although effective, counting-by-density is still challenged with scale variations induced by perspective distortion. To address this problem, most methods [82, 10, 45] employ deep Convolutional Neural Network (CNN) for exploiting multi-scale features to perform density estimation in multi-scaled scenes. In particular, different-sized filters are arranged in parallel in multiple columns to capture multi-scale features for accurate counting in [82, 49, 58], while in [10, 29, 31], different filters are grouped into blocks and then stacked sequentially in one column. At the heart of these solutions, multi-scale capability originates from the compositional nature of CNN [7, 73, 26], where convolutions with various receptive fields are composed hierarchically by hand. However, these manual designs demand prohibitive expert-efforts.

We therefore develop a Neural Architecture Search (NAS) [84, 53] based approach to automatically discover the multi-scale counting-by-density models. NAS is enabled by the compositional nature of CNN and guided by human expertise in designing task-specific search space and strategies. For vision tasks, NAS blooms with image-level classification [85, 39, 52, 51], where novel architectures are found to progressively transform spatial details to semantically deep features. Counting-by-density is, however, a pixel-level task that requires spatial preserving architectures with refrained down-sampling strides. Accordingly, the successes of NAS in image classification are not immediately transferable to crowd counting. Although attempts have been made to deploy NAS in image segmentation for pixel-level classifications [13, 38, 47], they are still not able to address counting-by-density, which is a pixel-level regression task with scale variations across the inputs.

In our NAS-Count, we propose a counting-oriented NAS framework with specific search strategy, search space and supervision method to develop our Automatic Multi-Scale Network (AMSNet). First, to achieve a fast search speed, we adopt a differential one-shot search strategy [41, 38, 77], in which architecture parameters are jointly optimized with gradient-based optimizer. Second, we employ a counting-specific two-level search space [59, 38]. On the micro-level, multi-scale cells are automatically explored to extract and fuse multi-scale features sufficiently. Pooling operations are limited to preserve spatial information and dilated convolutions are utilized instead for receptive field enlargement. On the macro-level, multi-path encoder-decoder architectures are searched to fuse multi-scale features from different cells and produce a high-quality density map.Fully-convolutional encoder-decoder is the architecture-of-choice for pixel-level tasks [54, 79, 10], and the multi-path variant can better aggregate features encoded at different scales [37, 31, 42]. However, previous differential one-shot search strategies [41, 77, 15] mainly concentrate on the single-path network and neglects the effect of feature aggregation, which cannot efficiently fuse multi-scale features from different stages and is not suitable for crowd counting task. In our work, the multi-path exploration in macro-level search can solve this is-



(a) (b)

Fig. 1: **(a):**An illustration of NAS-Count with the AMSNet architecture and SPPLoss supervision, all searched cells are outlined in black. Given $W \times W \times C$ ($C = 3$) inputs, the output dimension of each extraction and fusion cell are marked accordingly. **(b):**Detailed illustrations of the best searched cells. The circled additive sign denotes element-wise or scalar additions.

sue. Third, in order to address the pixel-level isolation problem [10, 36] of the traditional mean square error (MSE) loss, we propose to search an efficient Scale Pyramid Pooling Loss (SPPLoss) to optimize AMSNet. Leveraging the pyramidal pooling architecture to enforce supervision with multi-scale structural information has been prove effective in crowd counting task [31, 27, 17]. However, its best internal components have not been explored well. Here, in our NAS-Count, we take a further step and automatically searched the best operation to extract multi-scale information in SPPLoss, which provides the more efficient supervision than manually designed one. By jointly searching AMSNet and SPPLoss, NAS-Count flexibly exploits multi-scale features and addresses the scale variation issue in counting-by-density. NAS-Count is illustrated in Figure 1(a).

Main contributions of NAS-Count includes:

- To our best knowledge, NAS-Count is the first attempt at introducing NAS for crowd counting, where a multi-scale architecture is automatically developed to address the scale variation issue.
- A counting-specific two-level search space is developed in NAS-Count, from which a multi-path encoder-decoder architecture (AMSNet) is discovered efficiently with a differentiable search strategy using stochastic gradient descent (SGD).
- A Scale Pyramid Pooling Loss (SPPLoss) is searched automatically to improve MSE supervision, which helps produce the higher-quality density map via optimizing structural information on multiple scales.
- By jointly searching AMSNet and SSPLoss, NAS-Count reports the best overall counting and density estimation performances on four challenging benchmarks, considerably surpassing other state-of-the-arts which all require demanding expert-involvement.

2 Related work

2.1 Crowd Counting Literature

Existing counting methods can be categorized into counting-by-detection [18, 21, 35, 69], counting-by-regression [11, 55, 28, 81, 33], and counting-by-density strategies. For comprehensive surveys in crowd counting, please refer to [56, 57, 63, 32]. The first strategy is vulnerable to occlusions due to the requirement of explicit detection. Counting-by-regression successfully avoids such requirement by directly regressing to a scalar count, but forfeits the ability to perceive the localization of crowds. The counting-by-density strategy, initially introduced in [34], counts the crowd by first estimating a density map using hand-crafted [34, 20] or deep CNN [82, 36, 70, 44] features, then summing over all pixel values in the map. Being a pixel-level regression task, CNN architectures deployed in counting-by-density methods tend to follow the encoder-decoder formulation. In order to handle scale variations with multi-scale features, single-column [10, 29, 71] and multi-column [82, 5, 49, 67] encoders have been used where different-sized convolution kernels are sequentially or parallelly arranged to extract features. For the decoder, hour-glass architecture with a single decoding path has been adopted [10, 29, 83], while a novel multi-path variant is gaining increasing attention for superior multi-scale feature aggregation [31, 42, 48, 47].

2.2 NAS Fundamentals

Although CNN have made great progress and achieved convincing performance in many computer vision tasks [24, 14, 25, 30], its inherent structure often relies on the manual design, which demands enormous manpower and time. NAS, aiming to automatically explore the best structure of the network, has received considerable attention in recent years. The general efforts of developing new NAS solutions focus on designing new search spaces and search strategies. For search space, existing methods can be categorized into searching the network (macro) space [53, 84], the cell (micro) space [85, 39, 52, 41, 50], or exploring such a two-level space [59, 38] jointly. The cell-based space search is the most popular where the ensemble of cells in networks is hand-engineered to reduce the exponential search space for fast computation. For search strategy, it is essentially an optimizer to find the best architecture that maximizes a targeted task-objective. Random search [4, 23], reinforcement learning [84, 85, 66, 8, 1], neuro-evolutionary algorithms [65, 53, 46, 40, 52, 72], and gradient-based methods [41, 74, 9] have been used to solve the optimization problem, but the first three suffer from prohibitive computation costs. Although many attempts have been made such as parameter sharing [50, 19, 8, 3], hierarchical search [39, 38], deploying proxy tasks with cheaper search space [85] and training procedures [2] to accelerate them, yet they are still far less efficient than gradient-based methods. Gradient-based NAS, represented by DARTS [41], follows the one-shot strategy [6] wherein a hyper-graph is established using differentiable architectural parameters. Based on the hyper-graph, an optimal sub-graph is explored within by solving a bi-level optimization with gradient-descent optimizers.

2.3 NAS Applications

NAS has shown great promise with discovered recurrent or convolutional neural networks in both sequential language modeling [64] and multi-level vision tasks. In computer vision, NAS has excelled at image-level classification tasks [85, 51, 39, 52], which is a customary starting-point for developing new classifiers outputting spatially coarsened labels. NAS was later extended to both bounding-box and pixel-level tasks, represented by object detection [22, 16, 76] and segmentation [13, 38, 47], where the search spaces are modified to better preserve the spatial information in the feature map. In [13] a pixel-level oriented search space and a random search NAS were introduced to the pixel-level segmentation task. In [47], a similar search space was adopted, but the authors employed a reinforcement learning based search method. Nonetheless, both two methods suffer from formidable computations and are orders of magnitude slower than NAS-Count. In [38], the authors searched a two-level search space with more efficient gradient-based method, yet it dedicates in solving the pixel-level classification in semantic segmentation, which still differs from the per-pixel regression in counting-by-density.

3 NAS-Count Methodology

NAS-Count efficiently searches a multi-scale encoder-decoder network, the Automatic Multi-Scale Network (AMSNet) as shown in Figure 1(a), in a counting-specific search space. It is then optimized with a jointly searched Scale Pyramid Pooling Loss (SPPLoss). The encoder and decoder in AMSNet consist of searched multi-scale feature extraction cells and multi-scale feature fusion cells, respectively, and SPPLoss deploys a two-stream pyramidal pooling architecture where the pooling cells are searched as well. By searching AMSNet and SPPLoss together, the operations searched in these two architectures can collaborate with each other to obtain the ideal multi-scale capability for addressing the scale-variation problem in crowd counting. NAS-Count details are discussed in the following subsections.

3.1 Automatic Multi-Scale Network

AMSNet is searched with the differential one-shot strategy in a two-level search space. To improve the search efficiency, NAS-Count adopts a continuous relaxation and partial channel connection as described in [77]. Differently, to alter the single-path formulation in [77], we utilize the macro-level search to explore a multi-path encoder-decoder formulation for sufficient multi-scale feature extraction and fusion.

AMSNet Encoder The encoder of AMSNet is composed of a set of multi-scale feature extraction cells. For the l-th cell in the encoder, it takes the outputs of previous two cells, feature maps x_{l-2} and x_{l-1}, as inputs and produces an output feature map x_l. We define each *cell* as a directed acyclic graph containing N_e *nodes*, *i.e.* x_e^i with $1 \leqslant i \leqslant N_e$, each represents a propagated feature map. We set N_e=7 containing two input nodes, four intermediate nodes, and one output node.

Each directed *edge* in a cell indicates a convolutional operation $o_e(*)$ performed between a pair of nodes, and $o_e(*)$ is searched from the search space O_e with 9 operations:

- 1×1 common convolution;
- 3×3, 5×5, 7×7 dilated convolution with rate 2;
- 3×3, 5×5, 7×7 depth-wise separable convolution;
- skip-connection;
- no-connection (zero);

For preserving spatial fidelity in the extracted features, extraction cell involves no down-sampling operations. To compensate for the receptive field enlargement, we utilize dilated convolutions to substitute for the normal ones. Besides, we adopt depth-wise separable convolutions to keep the searched architecture parameter-friendly. Skip connections instantiate the residual learning scheme, which helps to improve multi-scale capacity as well as enhance gradient flows during back-propagation.

Within each cell, a specific intermediate node x_e^m is connected to all previous nodes x_e^1, x_e^2 $\cdots$, x_e^{m-1}. Edges $o_e^{n,m}(*)$ are established between every pair of connected-nodes n and m, forming a densely-connected hyper-graph. On a given edge $o_e^{n,m}(*)$ in the graph, following the continuously-relaxed differentiable search as discussed in [41], its associated operation is defined as a summation of all possible operations weighted by the architectural parameter α_e:

$$o_e^{n,m}\left(x_e^n; S\right) = \sum_i \sigma(\alpha_e^{n,m,i}) \cdot o_e^i\left(S \cdot x_e^n\right) + (1 - S) \cdot x_e^n, \tag{1}$$

in the above equation, $\sigma(*)$ is a softmax function and $i = 9$ indicates the volume of the micro-level search space. Vector S is applied to perform a channel-wise sampling on x_e^n, where $1/K$ channels are randomly selected to improve the search efficiency. K is set to 4 as proposed in [77]. $\alpha_e^{n,m}$ is a learnable parameter denoting the importance of each operation on an edge $o_e^{n,m}(*)$.

In addition, each edge is also associated with another architecture parameter $\beta_e^{n,m}$ which indicates its importance. Accordingly, an intermediate node x_e^m is computed as a weighted sum of all edges connected to it:

$$x_e^m = \sum_{n<m} \sigma(\beta_e^{n,m}) \cdot o_e^{n,m}\left(x_e^n; S\right) \tag{2}$$

here, n includes all previous nodes in the cell. The output of the cell is a concatenation of all its intermediate nodes. The cell architecture is determined by two architectural parameters α_e and β_e, which are jointly optimized with the weights of convolutions through a bi-level optimization. For details please refer to [41]. To recover a deterministic architecture from continuous relaxation, the most important edges as well as their associated operations are determined by computing *argmax* on the product of $\sigma\left(\beta_e\right)$ and corresponding $\sigma\left(\alpha_e\right)$.

In the encoder, we apply a 1×1 convolution to preliminary encode the input image into a $\frac{C}{4}$ channel feature map. Afterwards, two 1×1 convolutions are implemented after the first and third extraction cells, each doubling the channel dimension of the features. Our searched extraction cell is normal cell that keeps the feature channel dimension unchanged. Spatially, we only reduce the feature resolution twice through two max pooling layers, aiming to preserve the spatial fidelity in the features, while double the channels before the two down-sampling operations. Additionally, within each extraction cell, an extra 1×1 convolution is attached to each input node, adjusting their feature channels to be one-fourth of the cell final output dimension.

AMSNet Decoder The decoder of AMSNet deploys a multi-scale feature fusion cell followed by an up-sampling module. We construct the hyper-graph of the fusion cell as inputting multiple features while outputting just one, therefore conforming to the aggregative nature of a decoder. The search in this hyper-graph is similar to that of the extraction cell. A fusion cell takes three encoder output feature maps as input, consisting of $N_f = 6$ nodes that include three input nodes, two intermediate nodes and one output node. After the relaxation as formulated in Eqa.1 and 2, the architecture of a fusion cell is determined by its associated architecture parameters α_d and β_d. By optimizing β_d on three edges connecting the decoder with three extraction cells in the encoder, NAS-Count fully explores the macro-level architecture of AMSNet, such that different single- or multi-path encoder-decoder formulations are automatically searched to discover the best feature aggregation for producing high-quality density maps. Through this macro-level search, we extend PC-DARTS from the single-path search strategy to a newly multi-path search strategy, which is more suitable for discovering a multi-scale network for crowd counting task.

As shown in Figure 1(a), M denotes the number of extraction cells in the encoder and C is the number of channels in the output of the last cell. To improve efficiency, we first employ a smaller proxy network, with M=6 and C=256, to search the cell architecture. Upon deployment, we enlarge the network to M=8 and C=512 for better performance. Through the multi-scale aggregation in the decoder, we obtain a feature map with 128 channels, which is then processed by an up-sampling module containing two 3×3 convolutions interleave with the nearest neighbor interpolation layers. The output of the up-sampling module is a single-channel density map with restored spatial resolution, which is then utilized in computing the SPPLoss.

3.2 Scale Pyramid Pooling Loss

The default loss function to optimize counting-by-density models is the per-pixel mean square error (MSE) loss. By supervising this L_2 difference between the estimated density map and corresponding ground-truth, one assumes strong pixel-level isolation, such that it fails to reflect structural differences in multi-scale regions [10, 36]. As motivated by the Atrous Spatial Pyramid Pooling (ASPP) module designed in [14], previous work [31] attempts to solve this problem by proposing a new supervision architecture where non-parametric pooling layers are stacked into a two-stream pyramid. We call this supervision as Scale Pyramid Pooling Loss (SPPLoss). As shown in Figure 1(b), after feeding the estimated map E and the ground-truth G into each stream, they are progressively coarsened and MSE losses are calculated on each level between the pooled maps. This is equivalent to computing the structural difference with increasing region-level receptive fields, and can therefore better supervise the pixel-level estimation model on different scales.

Instead of setting the pooling layers manually as in [31], NAS-Count searches the most effective SPPLoss architecture jointly with AMSNet. In this way, the multi-scale capability composed in both architecture can better collaborate to resolve the scale variation problem in counting-by-density. Specifically, each stream in SPPLoss deploys N_l=4 cascaded nodes. Among them, one input node is the predicted density map (or the given ground-truth). The other three nodes are produced through three cascaded searched pooling layers. The search space for operation O_l performed on each edge contains six different pooling layers including:

- 2×2, 4×4, 6×6 max pooling layer with stride 2;
- 2×2, 4×4, 6×6 average pooling layer with stride 2;

The search for SPPLoss adopts the similar differentiable strategy as detailed in Section 3.1. Notably, as SPPLoss is inherently a pyramid, its macro-level search space takes a cascaded form instead of a densely-connected hyper-graph. Accordingly, we only need to optimize the operation-wise architecture parameter α_s as follows:

$$o_s^{n,m}\left(x_s^n\right) = \sum_i \sigma\left(\alpha_s^{n,m,i}\right) \cdot o_s^i\left(x_s^n\right) \tag{3}$$

i indicates 6 different pooling operations, and x_s^n represents an estimated map E or ground-truth G in specific level. Since both of them only have one channel, we thus do not apply partial channel connections (*i.e.* set K equals to 1). The same cascaded architecture is shared in both streams of SPPLoss. Using the best searched architecture as depicted in Figure 1(b), SPPLoss is computed as:

$$L_{SPP} = \sum_n \frac{1}{N^l}\left\|\phi^l(E) - \phi^l(G)\right\|_2^2 \tag{4}$$

N^l denotes the number of pixels in the map, $\phi^l(*)$ indicates the searched pooling operation, superscript l is the layer index ranging from 0 to 3. $l = 0$ is the special case where MSE loss is computed directly between E and G.

4 Experiments

4.1 Implementation Details

The original annotations provided by the datasets are coordinates pinpointing the location of each individual in the crowd. To soften these hard regression labels for better convergence, we apply a normalized 2D Gaussian filter to convert coordinate map into density map, on which each individual is represented by a Gaussian response with radius equals to 15 pixels [71].

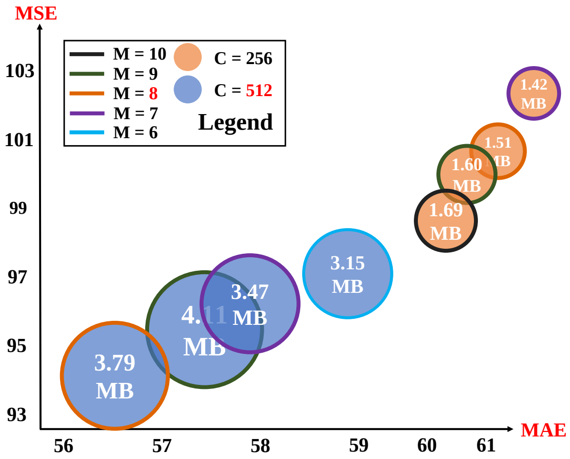


Fig. 2: Illustrated hyper-parameter analysis. M is the number of extraction cells, C denotes the channels of feature map generated by the last extraction cell. Bottom left corner indicates superior counting result and the number in the circle indicates the parameter size of each model. The best hyper-parameters are colored with red in the legend.

Architecture Search The architecture of AMSNet and SPPLoss, *i.e.* their corresponding architecture parameters $\alpha_{e,d,s}$ and $\beta_{e,d}$, are jointly searched on the UCF-QNRF [29] training set. We choose to perform search on this dataset as it has the most challenging scenes with large crowd counts and density variations, and the search costs approximately 21 TITAN Xp GPU hours. Benefiting from the continuous relaxation, we optimize all architecture parameters and network weights w jointly using gradient descent. Specifically, the first-order optimization proposed in [41] is adopted, upon which w and α, β are optimized alternatively. For architecture parameters, we set the learning rate to be 6e-4 with weight decay of 1e-3. We follow the implementation as in [77, 38], where a warm-up training for network weights is first conducted for 40 epochs and stops the search early at 80 epochs. For training the network weights, we use a cosine learning rate that decays from 0.001 to 0.0004, and weight decay 1e-4. Data augmentation including random-scale sampling, random flip and random rotation are conducted to alleviate overfitting.

Architecture Training After the architectures of AMSNet and SPPLoss are determined by searching on the UCF-QNRF dataset, we re-train the network weights w from scratch on each dataset respectively. We re-initialize the weights with Xavier initialization, and employ Adam with initial learning rate set to 1e-3. This learning rate is decayed by 0.8 every 15K iterations.

Architecture Evaluation Upon deployment, we directly feed the whole image into AMSNet, aiming to obtain high-quality density maps free from boundary artifacts. In counting-by-density, the crowd count on an estimated density map equals to the summation of all pixels. To evaluate the counting performance, we

Table 1: Estimation errors on the ShanghaiTech. The best performance is colored red and the second best is colored blue.

Method	ShanghaiTech Part_A		ShanghaiTech Part_B	
	MAE↓	MSE↓	MAE↓	MSE↓
MCNN [82]	110.2	173.2	26.4	41.3
CSRNet [36]	68.2	115.0	10.6	16.0
SANet [10]	67.0	104.5	8.4	13.6
CFF [61]	65.2	109.4	7.2	12.2
TEDNet [31]	64.2	109.1	8.2	12.8
SPN+L2SM [75]	64.2	98.4	7.2	11.1
ANF [80]	63.9	99.4	8.3	13.2
PACNN+ [60]	62.4	102.0	7.6	11.8
CAN [44]	62.3	100.0	7.8	12.2
SPANet [17]	59.4	92.5	6.5	9.9
PGCNet [78]	57.0	86.0	8.8	13.7
AMSNet	56.7	93.4	6.7	10.2

Table 2: Estimation errors on the UCF_CC_50 and the UCF-QNRF datasets. The best performance is colored red and the second best is colored blue.

Method	UCF_CC_50		UCF-QNRF	
	MAE↓	MSE↓	MAE↓	MSE↓
Zhang *et al.* [81]	467.0	498.5	-	-
MCNN [82]	377.6	509.1	277	426
CP-CNN [62]	295.8	320.9	-	-
CSRNet [36]	266.1	397.5	-	-
SANet [10]	258.4	334.9	-	-
TEDNet [31]	249.4	354.5	113	188
ANF [80]	250.2	340.0	110	174
PACNN+ [60]	241.7	320.7	-	-
CAN [44]	212.2	243.7	107	183
CFF [61]	-	-	93.8	146.5
SPN+L2SM [75]	188.4	315.3	104.7	173.6
AMSNet	208.4	297.3	101.8	163.2

follow the previous work and employ the widely used mean average error (MAE) and the mean squared error (MSE) metrics. Additionally, we also utilize the PSNR (Peak Signal-to-Noise Ratio) and SSIM (Structural Similarity in Image) metrics to evaluate density map quality [62].

4.2 Search Result Analysis

The best searched multi-scale feature extraction and fusion cells, as well as the SPPLoss architecture are illustrated in Figure 1(b). As shown, extraction cell maintains the spatial and channel dimensions unchanged (1×1 convolutions are employed to manipulate the channel dimensions in the cells). The extraction cell primarily exploits dilated convolutions over normal ones, conforming to the fact that in the absence of heavy down-samplings, pixel-level models rely on dilations

Table 3: The MAE comparison on WorldExpo'10. The best performance is colored red and second best is colored blue.

Method	S1	S2	S3	S4	S5	Ave.
SANet [10]	2.6	13.2	9.0	13.3	3.0	8.2
CAN [44]	2.9	12.0	10.0	7.9	4.3	7.4
DSSIN [43]	1.6	9.5	9.5	10.4	2.5	6.7
ECAN [44]	2.4	9.4	8.8	11.2	4.0	7.2
TEDNet [31]	2.3	10.1	11.3	13.8	2.6	8.0
AT-CSRNet [83]	1.8	13.7	9.2	10.4	3.7	7.8
ADMG [68]	4.0	18.1	7.2	12.3	5.7	9.5
AMSNet	1.6	8.8	10.8	10.4	2.5	6.8

to enlarge receptive fields. Furthermore, different kernel sizes are employed in the extraction cell, showing its multi-scale capability in addressing scale variations. By taking in three encoded features and generating one output feature, the fusion cell constitutes a multi-path decoding hierarchy, wherein primarily non-dilated convolutions with smaller kernels are selected to aggregate features more precisely and parameter-friendly.

4.3 Ablation Study on Searched Architectures

For ablation purposes, we employ the architecture proposed in [10] as the baseline encoder (composed of four inception-like blocks). Additionally, to better elaborate the effectiveness of the search process, we also employ the backbone searched on ImageNet in [77] to compose the classification encoder (For the consideration of computation cost and fair comparison, we totally set 8 cells in encoder, which is the same in our AMSNet). The baseline decoder cascades two 3×3 convolutions interleaved with nearest-neighbor interpolation layers. The normal MSE loss is utilized as baseline supervision. By comparing different modules with its baseline, the ablation study results on the ShanghaiTech Part_A dataset are reported in Table 4. This table is partitioned into three groups, and each row indicates a specific configuration. The MAE and PSNR metrics are used to show the counting accuracy and density map quality.

Architectures in the first two groups (five rows) are optimized with the normal MSE loss. As shown, compared to the baseline, the searched AMSNet encoder improves counting accuracy and density map quality by 12.7% and 9.7%, while the searched decoder brings 9.7% and 5.1% improvements respectively. Meanwhile, compared to the classification encoder, AMSNet encoder also improves the performance by 11.1% and 9.1% in MAE and PSNR, which indicates we obtain a more powerful backbone for multi-scale feature extraction through the search process. In the third group, AMSNet is supervised by different loss functions to demonstrate their efficacy. The Spatial Abstraction Loss (SAL) proposed in [31] adopts a hand-designed pyramidal architecture, which surpasses the normal MSE supervision on both counting and density estimation performance.

Table 4: Ablation study results. Best performance is bolded, and arrows indicate the favorable directions of the metric values.

Configurations			MAE↓	PSNR↑
Encoder Architecture	1	Baseline Encoder Baseline Decoder	69.1	23.54
	2	Classification Encoder Baseline Decoder	67.8	23.67
	3	AMSNet Encoder Baseline Decoder	**60.3**	**25.82**
Decoder Architecture	1	Baseline Encoder Baseline Decoder	69.1	23.54
	4	Baseline Encoder AMSNet Decoder	**62.4**	**24.75**
Supervision	5	AMSNet + MSE	58.5	26.17
	6	AMSNet + SAL	57.6	26.62
	7	AMSNet + SPPLoss	**56.7**	**27.03**

Table 5: Model size and performance comparison among state-of-the-art counting methods on the ShanghaiTech Part_A.

Method	MAE↓	PSNR↑	SSIM↑	Size
MCNN [82]	110.2	21.4	0.52	**0.13MB**
Switch-CNN [58]	90.4	-	-	15.11MB
CP-CNN [62]	73.6	21.72	0.72	68.4MB
CSRNet [36]	68.2	23.79	0.76	16.26MB
SANet [10]	67.0	-	-	0.91MB
TEDNet [31]	64.2	25.88	0.83	1.63MB
ANF [80]	63.9	24.1	0.78	7.9MB
AMSNet	**56.7**	**27.03**	**0.89**	3.79MB
AMSNet_light	61.3	26.18	0.85	1.51MB

These improvements are further enhanced by deploying SPPLoss, showing that the searched pyramid benefits counting and density estimation by supervising multi-scale structural information.

Furthermore, we also compare AMSNet decoder with some existing multi-path decoder to show the ability of our macro-level search in discovering an efficient feature aggregation configuration. These experiments are elaborated in detail in the supplementary material.

4.4 Hyper-parameter Study

The size and performance of AMSNet are largely dependent on two hyper-parameter M and C, each denoting the number of extraction cell and its output channel dimension. As illustrated in Figure 2, $M = 8$ and $C = 512$ render the best counting performance, but populate AMSNet with 3.79MB parameters. When decreasing C to 256, the size of AMSNet also shrinks dramatically, but at the expense of decreased accuracy. Nevertheless, $M = 8$ still produces the best

MAE in this case. As a result, we configure our AMSNet with $M = 8, C = 512$, and also establish an AMSNet_light with $M = 8, C = 256$ in the experiment.

We compare the counting accuracy and density map quality of both AMSNet and AMSNet_light with other state-of-the-art counting methods in Table 5. As shown, AMSNet reports the best MAE and PSNR overall, while being heavier than three other methods. AMSNet_light, on the other hand, is the third most light model and achieves the best performance with the exception of AMSNet.

4.5 Performance and Comparison

We compare the counting-by-density performance of NAS-Count with other state-of-the-art methods on four challenging datasets, ShanghaiTech [82], WorldExpo'10 [81], UCF_CC_50 [28] and UCF-QNRF [29]. In particular, the counting accuracy comparison is reported in Tables 1, 2 and 3, while the density map quality result is shown in Table 5.

Counting Accuracy The ShanghaiTech is composed of Part_A and Part_B with in total of 1198 images. It is one of the largest and most widely used datasets in crowd counting. As shown in Table 1, AMSNet achieves the state-of-the-art performance in terms of both MAE and MSE. On Part_A, we achieve the best MAE and the competitive MSE. On Part_ B, we report the second best MAE and MSE, which are only a little inferior to [17].

The UCF_CC_50 dataset contains 50 images of varying resolutions and densities. In consideration of sample scarcity, we follow the standard protocol [28] and use 5-fold cross-validation to evaluate method performance. As shown in Table 2, we achieve the second best MAE and MSE. It is worth mentioning that, although our MAE is a higher than SPN+L2SM [75], our MSE is obviously better than it. Meanwhile, our MAE is also superior to CAN [44], which is the only current method achieves a lower MSE than our AMSNet. Therefore, AMSNet produces the best performance when we comprehensively consider both MAE and MSE together.

The UCF-QNRF dataset introduced by Idress *et al.* [29] has images with the highest crowd counts and largest density variation, ranging from 49 to 12865 people per image. These characteristics make UCF-QNRF extremely challenging for counting models. As shown in Table 2, we achieve the second best performance in terms of both MAE and MSE on this dataset.

The WorldExpo'10 dataset [81] contains 3980 images covering 108 different scenes. As shown in Table 3, AMSNet achieves the second lowest average MAE over five scenes, and also performs the best on the three scenes individually.

It is worth noting that although we do not produce the best counting accuracy on every dataset. Our AMSNet is the only method that achieves the top-two performance on the four employed datasets simultaneously. In the other word, AMSNet performs best when we comprehensively consider the four datasets.

Density Map Quality As shown in Table 5, we employ PSNR and SSIM indices to compare the quality of density maps estimated by different methods. AMSNet performs the best on both indices, outperforming the second best by

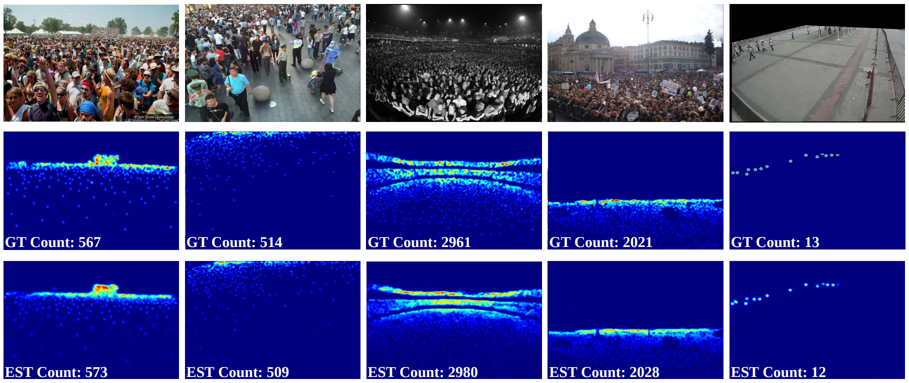


Fig. 3: An illustration of generated density maps on ShanghaiTech Part_A, ShanghaiTech Part_B, UCF_50_CC, UCF-QNRF and WorldExpo'10 respectively. The first row shows the input images, the second and third depict the ground truth and estimated density maps.

4.4% and 7.2% respectively. Notably, even by deploying AMSNet_light which is the third lightest model, we still generate the most high-quality density map. We further showcase more density maps generated by AMSNet on all employed datasets in Figure 3.

5 Conclusion

NAS-Count is the first endeavor introducing neural architecture search into counting-by-density. In this paper, we extend PC-DARTS [77] to a counting-specific two-level search space, in which micro- and macro-level search are employed to explore a multi-path encoder-decoder network, AMSNet, as well as the SPPLoss. Specifically, AMSNet employs a novel composition of multi-scale feature extraction and fusion cells. SPPLoss automatically searches a scale pyramid architecture to extend normal MSE loss, which helps to supervise structural information in the density map at multiple scales. By jointly searching AMSNet and SPPLoss end-to-end, NAS-Count surpasses tedious hand-designing efforts by achieving a multi-scale model automatically with less than 1 GPU day, and demonstrates overall the best performance on four challenging datasets.

Acknowledgment

This paper was supported by the National Natural Science Foundation of China (NSFC) under grant 91738301, and the National Key Scientific Instrument and Equipment Development Project under Grant 61827901.

References

1. Baker, B., Gupta, O., Naik, N., Raskar, R.: Designing neural network architectures using reinforcement learning. arXiv preprint arXiv:1611.02167 (2016)
2. Baker, B., Gupta, O., Raskar, R., Naik, N.: Accelerating neural architecture search using performance prediction. arXiv preprint arXiv:1705.10823 (2017)
3. Bender, G., Kindermans, P.J., Zoph, B., Vasudevan, V., Le, Q.: Understanding and simplifying one-shot architecture search. In: International Conference on Machine Learning. pp. 549–558 (2018)
4. Bergstra, J., Bengio, Y.: Random search for hyper-parameter optimization. Journal of Machine Learning Research **13**(Feb), 281–305 (2012)
5. Boominathan, L., Kruthiventi, S.S., Babu, R.V.: Crowdnet: A deep convolutional network for dense crowd counting. In: Proceedings of the 2016 ACM on Multimedia Conference. pp. 640–644. ACM (2016)
6. Brock, A., Lim, T., Ritchie, J.M., Weston, N.: Smash: one-shot model architecture search through hypernetworks. arXiv preprint arXiv:1708.05344 (2017)
7. Bronstein, M.M., Bruna, J., LeCun, Y., Szlam, A., Vandergheynst, P.: Geometric deep learning: going beyond euclidean data. IEEE Signal Processing Magazine **34**(4), 18–42 (2017)
8. Cai, H., Chen, T., Zhang, W., Yu, Y., Wang, J.: Efficient architecture search by network transformation. In: Proceedings of the AAAI Conference on Artificial Intelligence (2018)
9. Cai, H., Zhu, L., Han, S.: Proxylessnas: Direct neural architecture search on target task and hardware. arXiv preprint arXiv:1812.00332 (2018)
10. Cao, X., Wang, Z., Zhao, Y., Su, F.: Scale aggregation network for accurate and efficient crowd counting. In: Proceedings of the European Conference on Computer Vision. pp. 734–750 (2018)
11. Chan, A.B., Vasconcelos, N.: Bayesian poisson regression for crowd counting. In: Proceedings of the International Conference on Computer Vision. pp. 545–551 (2009)
12. Chen, K., Loy, C.C., Gong, S., Xiang, T.: Feature mining for localised crowd counting. In: Proceedings of the British Machine Vision Conference. vol. 1, p. 3 (2012)
13. Chen, L.C., Collins, M., Zhu, Y., Papandreou, G., Zoph, B., Schroff, F., Adam, H., Shlens, J.: Searching for efficient multi-scale architectures for dense image prediction. In: Proceedings of the Advances in Neural Information Processing Systems. pp. 8699–8710 (2018)
14. Chen, L.C., Papandreou, G., Kokkinos, I., Murphy, K., Yuille, A.L.: Deeplab: Semantic image segmentation with deep convolutional nets, atrous convolution, and fully connected crfs. IEEE Transactions on Pattern Analysis and Machine Intelligence **40**(4), 834–848 (2017)
15. Chen, X., Xie, L., Wu, J., Tian, Q.: Progressive differentiable architecture search: Bridging the depth gap between search and evaluation. arXiv preprint arXiv:1904.12760 (2019)
16. Chen, Y., Yang, T., Zhang, X., Meng, G., Pan, C., Sun, J.: Detnas: Neural architecture search on object detection. arXiv preprint arXiv:1903.10979 (2019)
17. Cheng, Z.Q., Li, J.X., Dai, Q., Wu, X., Hauptmann, A.G.: Learning spatial awareness to improve crowd counting. In: Proceedings of the IEEE International Conference on Computer Vision. pp. 6152–6161 (2019)
18. Dollar, P., Wojek, C., Schiele, B., Perona, P.: Pedestrian detection: An evaluation of the state of the art. IEEE Transactions on Pattern Analysis and Machine Intelligence **34**(4), 743–761 (2012)

19. Elsken, T., Metzen, J.H., Hutter, F.: Simple and efficient architecture search for convolutional neural networks. arXiv preprint arXiv:1711.04528 (2017)
20. Fiaschi, L., Köthe, U., Nair, R., Hamprecht, F.A.: Learning to count with regression forest and structured labels. In: Proceedings of the International Conference on Pattern Recognition. pp. 2685–2688. IEEE (2012)
21. Ge, W., Collins, R.T.: Marked point processes for crowd counting. In: Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition. pp. 2913–2920 (2009)
22. Ghiasi, G., Lin, T.Y., Le, Q.V.: Nas-fpn: Learning scalable feature pyramid architecture for object detection. In: Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition. pp. 7036–7045 (2019)
23. Golovin, D., Solnik, B., Moitra, S., Kochanski, G., Karro, J., Sculley, D.: Google vizier: A service for black-box optimization. In: Proceedings of the 23rd ACM SIGKDD International Conference on Knowledge Discovery and Data Mining. pp. 1487–1495. ACM (2017)
24. He, K., Zhang, X., Ren, S., Sun, J.: Deep residual learning for image recognition. In: Proceedings of the IEEE conference on computer vision and pattern recognition. pp. 770–778 (2016)
25. Hu, J., Zhu, E., Wang, S., Wang, S., Liu, X., Yin, J.: Two-stage unsupervised video anomaly detection using low-rank based unsupervised one-class learning with ridge regression. In: 2019 International Joint Conference on Neural Networks (IJCNN). pp. 1–8. IEEE (2019)
26. Hu, Y., Yang, Y., Zhang, J., Cao, X., Zhen, X.: Attentional kernel encoding networks for fine-grained visual categorization. IEEE Transactions on Circuits and Systems for Video Technology (2020)
27. Huang, S., Li, X., Cheng, Z.Q., Zhang, Z., Hauptmann, A.: Stacked pooling: Improving crowd counting by boosting scale invariance. arXiv preprint arXiv:1808.07456 (2018)
28. Idrees, H., Saleemi, I., Seibert, C., Shah, M.: Multi-source multi-scale counting in extremely dense crowd images. In: Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition. pp. 2547–2554 (2013)
29. Idrees, H., Tayyab, M., Athrey, K., Zhang, D., Al-Maadeed, S., Rajpoot, N., Shah, M.: Composition loss for counting, density map estimation and localization in dense crowds. arXiv preprint arXiv:1808.01050 (2018)
30. Jiang, X., Li, P., Zhen, X., Cao, X.: Model-free tracking with deep appearance and motion features integration. In: 2019 IEEE Winter Conference on Applications of Computer Vision (WACV). pp. 101–110. IEEE (2019)
31. Jiang, X., Xiao, Z., Zhang, B., Zhen, X., Cao, X., Doermann, D., Shao, L.: Crowd counting and density estimation by trellis encoder-decoder networks. In: Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition. pp. 6133–6142 (2019)
32. Kang, D., Ma, Z., Chan, A.B.: Beyond counting: comparisons of density maps for crowd analysis tasks-counting, detection, and tracking. IEEE Transactions on Circuits and Systems for Video Technology (2018)
33. Kumagai, S., Hotta, K., Kurita, T.: Mixture of counting cnns: Adaptive integration of cnns specialized to specific appearance for crowd counting. arXiv preprint arXiv:1703.09393 (2017)
34. Lempitsky, V., Zisserman, A.: Learning to count objects in images. In: Proceedings of the Advances in Neural Information Processing Systems. pp. 1324–1332 (2010)

35. Li, M., Zhang, Z., Huang, K., Tan, T.: Estimating the number of people in crowded scenes by mid based foreground segmentation and head-shoulder detection. In: Proceedings of the International Conference on Pattern Recognition. pp. 1–4. IEEE (2008)
36. Li, Y., Zhang, X., Chen, D.: Csrnet: Dilated convolutional neural networks for understanding the highly congested scenes. In: Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition. pp. 1091–1100 (2018)
37. Lin, G., Milan, A., Shen, C., Reid, I.D.: Refinenet: Multi-path refinement networks for high-resolution semantic segmentation. In: Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition. vol. 1, p. 5 (2017)
38. Liu, C., Chen, L.C., Schroff, F., Adam, H., Hua, W., Yuille, A.L., Fei-Fei, L.: Auto-deeplab: Hierarchical neural architecture search for semantic image segmentation. In: Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition. pp. 82–92 (2019)
39. Liu, C., Zoph, B., Neumann, M., Shlens, J., Hua, W., Li, L.J., Fei-Fei, L., Yuille, A., Huang, J., Murphy, K.: Progressive neural architecture search. In: Proceedings of the European Conference on Computer Vision. pp. 19–34 (2018)
40. Liu, H., Simonyan, K., Vinyals, O., Fernando, C., Kavukcuoglu, K.: Hierarchical representations for efficient architecture search. arXiv preprint arXiv:1711.00436 (2017)
41. Liu, H., Simonyan, K., Yang, Y.: Darts: Differentiable architecture search. arXiv preprint arXiv:1806.09055 (2018)
42. Liu, L., Qiu, Z., Li, G., Liu, S., Ouyang, W., Lin, L.: Crowd counting with deep structured scale integration network. In: Proceedings of the International Conference on Computer Vision. pp. 1774–1783 (2019)
43. Liu, L., Qiu, Z., Li, G., Liu, S., Ouyang, W., Lin, L.: Crowd counting with deep structured scale integration network. In: Proceedings of the IEEE International Conference on Computer Vision. pp. 1774–1783 (2019)
44. Liu, W., Salzmann, M., Fua, P.: Context-aware crowd counting. In: Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition (June 2019)
45. Liu, X., van de Weijer, J., Bagdanov, A.D.: Leveraging unlabeled data for crowd counting by learning to rank. arXiv preprint arXiv:1803.03095 (2018)
46. Miikkulainen, R., Liang, J., Meyerson, E., Rawal, A., Fink, D., Francon, O., Raju, B., Shahrzad, H., Navruzyan, A., Duffy, N., et al.: Evolving deep neural networks. In: Artificial Intelligence in the Age of Neural Networks and Brain Computing, pp. 293–312. Elsevier (2019)
47. Nekrasov, V., Chen, H., Shen, C., Reid, I.: Fast neural architecture search of compact semantic segmentation models via auxiliary cells. In: Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition. pp. 9126–9135 (2019)
48. Nekrasov, V., Shen, C., Reid, I.: Light-weight refinenet for real-time semantic segmentation. arXiv preprint arXiv:1810.03272 (2018)
49. Onoro-Rubio, D., López-Sastre, R.J.: Towards perspective-free object counting with deep learning. In: Proceedings of the European Conference on Computer Vision. pp. 615–629. Springer (2016)
50. Pham, H., Guan, M.Y., Zoph, B., Le, Q.V., Dean, J.: Efficient neural architecture search via parameter sharing. arXiv preprint arXiv:1802.03268 (2018)
51. Real, E., Aggarwal, A., Huang, Y., Le, Q.: Aging evolution for image classifier architecture search. In: Proceedings of the AAAI Conference on Artificial Intelligence (2019)

52. Real, E., Aggarwal, A., Huang, Y., Le, Q.V.: Regularized evolution for image classifier architecture search. In: Proceedings of the AAAI Conference on Artificial Intelligence. vol. 33, pp. 4780–4789 (2019)
53. Real, E., Moore, S., Selle, A., Saxena, S., Suematsu, Y.L., Tan, J., Le, Q.V., Kurakin, A.: Large-scale evolution of image classifiers. In: Proceedings of the 34th International Conference on Machine Learning-Volume 70. pp. 2902–2911. JMLR. org (2017)
54. Ronneberger, O., Fischer, P., Brox, T.: U-net: Convolutional networks for biomedical image segmentation. In: Proceedings of the International Conference on Medical Image Computing and Computer-assisted Intervention. pp. 234–241. Springer (2015)
55. Ryan, D., Denman, S., Fookes, C., Sridharan, S.: Crowd counting using multiple local features. In: Digital Image Computing: Techniques and Applications, 2009. DICTA'09. pp. 81–88. IEEE (2009)
56. Ryan, D., Denman, S., Sridharan, S., Fookes, C.: An evaluation of crowd counting methods, features and regression models. Computer Vision and Image Understanding **130**, 1–17 (2015)
57. Saleh, S.A.M., Suandi, S.A., Ibrahim, H.: Recent survey on crowd density estimation and counting for visual surveillance. Engineering Applications of Artificial Intelligence **41**, 103–114 (2015)
58. Sam, D.B., Surya, S., Babu, R.V.: Switching convolutional neural network for crowd counting. In: Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition. vol. 1, p. 6 (2017)
59. Saxena, S., Verbeek, J.: Convolutional neural fabrics. In: Proceedings of the Advances in Neural Information Processing Systems. pp. 4053–4061 (2016)
60. Shi, M., Yang, Z., Xu, C., Chen, Q.: Revisiting perspective information for efficient crowd counting. In: Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition. pp. 7279–7288 (2019)
61. Shi, Z., Mettes, P., Snoek, C.G.: Counting with focus for free. arXiv preprint arXiv:1903.12206 (2019)
62. Sindagi, V.A., Patel, V.M.: Generating high-quality crowd density maps using contextual pyramid cnns. In: Proceedings of the International Conference on Computer Vision. pp. 1879–1888. IEEE (2017)
63. Sindagi, V.A., Patel, V.M.: A survey of recent advances in cnn-based single image crowd counting and density estimation. Pattern Recognition Letters **107**, 3–16 (2018)
64. So, D.R., Liang, C., Le, Q.V.: The evolved transformer. arXiv preprint arXiv:1901.11117 (2019)
65. Stanley, K.O., Miikkulainen, R.: Evolving neural networks through augmenting topologies. Evolutionary Computation **10**(2), 99–127 (2002)
66. Tan, M., Chen, B., Pang, R., Vasudevan, V., Sandler, M., Howard, A., Le, Q.V.: Mnasnet: Platform-aware neural architecture search for mobile. In: Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition. pp. 2820–2828 (2019)
67. Walach, E., Wolf, L.: Learning to count with cnn boosting. In: Proceedings of the European Conference on Computer Vision. pp. 660–676. Springer (2016)
68. Wan, J., Chan, A.: Adaptive density map generation for crowd counting. In: Proceedings of the IEEE International Conference on Computer Vision. pp. 1130–1139 (2019)

69. Wang, M., Wang, X.: Automatic adaptation of a generic pedestrian detector to a specific traffic scene. In: Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition. pp. 3401–3408 (2011)
70. Wang, Q., Gao, J., Lin, W., Yuan, Y.: Learning from synthetic data for crowd counting in the wild. In: Proceedings of the The IEEE Conference on Computer Vision and Pattern Recognition (June 2019)
71. Wang, Z., Xiao, Z., Xie, K., Qiu, Q., Zhen, X., Cao, X.: In defense of single-column networks for crowd counting. arXiv preprint arXiv:1808.06133 (2018)
72. Xie, L., Yuille, A.: Genetic cnn. In: Proceedings of the International Conference on Computer Vision. pp. 1379–1388 (2017)
73. Xie, S., Kirillov, A., Girshick, R., He, K.: Exploring randomly wired neural networks for image recognition. arXiv preprint arXiv:1904.01569 (2019)
74. Xie, S., Zheng, H., Liu, C., Lin, L.: Snas: stochastic neural architecture search. arXiv preprint arXiv:1812.09926 (2018)
75. Xu, C., Qiu, K., Fu, J., Bai, S., Xu, Y., Bai, X.: Learn to scale: Generating multipolar normalized density maps for crowd counting. In: Proceedings of the IEEE International Conference on Computer Vision. pp. 8382–8390 (2019)
76. Xu, H., Yao, L., Zhang, W., Liang, X., Li, Z.: Auto-fpn: Automatic network architecture adaptation for object detection beyond classification. In: Proceedings of the International Conference on Computer Vision. pp. 6649–6658 (2019)
77. Xu, Y., Xie, L., Zhang, X., Chen, X., Qi, G.J., Tian, Q., Xiong, H.: Pc-darts: Partial channel connections for memory-efficient differentiable architecture search. arXiv preprint arXiv:1907.05737 (2019)
78. Yan, Z., Yuan, Y., Zuo, W., Tan, X., Wang, Y., Wen, S., Ding, E.: Perspective-guided convolution networks for crowd counting. In: Proceedings of the IEEE International Conference on Computer Vision. pp. 952–961 (2019)
79. Yang, J., Liu, Q., Zhang, K.: Stacked hourglass network for robust facial landmark localisation. In: Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition Workshops. pp. 2025–2033. IEEE (2017)
80. Zhang, A., Yue, L., Shen, J., Zhu, F., Zhen, X., Cao, X., Shao, L.: Attentional neural fields for crowd counting. In: Proceedings of the International Conference on Computer Vision (October 2019)
81. Zhang, C., Li, H., Wang, X., Yang, X.: Cross-scene crowd counting via deep convolutional neural networks. In: Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition. pp. 833–841 (2015)
82. Zhang, Y., Zhou, D., Chen, S., Gao, S., Ma, Y.: 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 (2016)
83. Zhao, M., Zhang, J., Zhang, C., Zhang, W.: Leveraging heterogeneous auxiliary tasks to assist crowd counting. In: Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition (June 2019)
84. Zoph, B., Le, Q.V.: Neural architecture search with reinforcement learning. arXiv preprint arXiv:1611.01578 (2016)
85. Zoph, B., Vasudevan, V., Shlens, J., Le, Q.V.: Learning transferable architectures for scalable image recognition. In: Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition. pp. 8697–8710 (2018)