PIS-NAS: Pyramid Information System-Based Neural Architecture Search for Object Detection

Chentian Wei[1,2] Ao Li[1] Lei Pu[3] Le Dong‡[1]
[1]School of Artificial Intelligence, Xidian University
[2]Institute for Network Sciences and Cyberspace, Tsinghua University
[3]Combat support College, Rocket Force University of Engineering

Background & Motivation

Manual design of network architectures is **time-consuming** and **labor-intensive**: Traditional methods require manual design of network structures and parameters, which requires extensive experience and trial and error, resulting in low efficiency.

Problems with existing work:

High computational resource consumption: Training-based NAS methods require large amounts of computing resources, limiting their feasibility in practical applications.

Limited search space: Zero-shot NAS methods usually focus on specific network structures, such as CNN or ResNet, and cannot explore a wider search space.

Lack of theoretical guarantees: Some existing NAS methods lack theoretical support, making it difficult to explain their design principles and performance advantages.

Overview

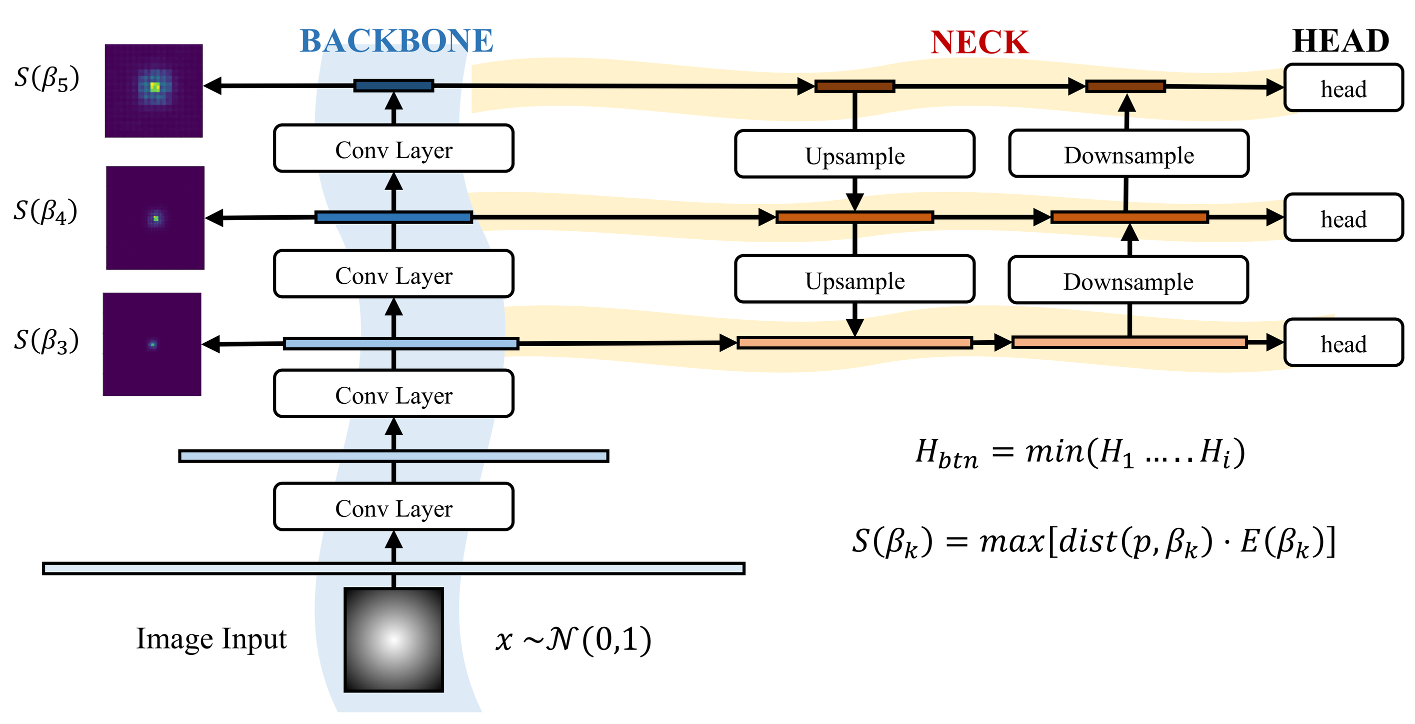


Figure 1. The architecture of proposed Pyramid Information System.

Contributions

- Zero-Shot Architecture Search Models NAS as constrained optimization via Pyramid Information System (PIS), eliminating training.
- Information-Theoretic Foundation Integrates **Pyramid Receptive Score** (information efficiency) and **Bottleneck Entropy** (transmission upper bound).
- Evolutionary Optimization Solves PIS constraints in <0.5 days, outperforming existing NAS methods.
- SOTA Performance Surpasses ResNet/YOLO on benchmarks under comparable computational budgets.

Welcome to explore our work and give us a ⭐ on GitHub!
Code is available at **https://github.com/ct-wei/PIS-NAS** 🚀

Design

We define the **Pyramid Receptive Score** which controls its efficiency, followed by a constraint named **Bottleneck Entropy**, which controls its upper bound of information transmission. In the end, the **final mathematical formulation** of PIS-NAS is presented.

- **Pyramid Receptive Score**
  The Pyramid Receptive Score(PR Score) quantifies the effective receptive field of convolutional neural networks (CNNs) across hierarchical feature maps.

$$G(x, \beta_k) = \mathrm{ReLU}\left(\sum_{i,j} \frac{\partial F(x)}{\partial c_{i,j}} \cdot \frac{\partial c_{i,j}}{\partial \beta_k}\right)$$
$$x \sim N(\mu, \sigma^2) \tag{1}$$

After normalization and thresholding, the pyramid receptive score of CNN is:

$$PRscore_i(\beta_k) \triangleq \max[dist(c_{0,0}^{in}, \beta_k) \cdot E(\beta_k)], \tag{2}$$

- **Bottleneck Entropy**
  Bottleneck Entropy quantifies the information transmission limit in convolutional neural networks (CNNs). It combines **Convolution Entropy** and **Image Entropy**:
  Convolution Entropy : Upper bound of differential entropy for a convolution layer(where W is the parameter matrix, and $x_{\text{in}} \sim \mathcal{N}(\mu, \sigma^2)$):

$$H_{\text{conv}} = \frac{1}{2} \log\left(2\pi e \cdot \sigma^2 (W \cdot x_{\text{in}})\right)$$

Image Entropy: Information from input resolution r^2:

$$H_{\text{img}} = \frac{1}{2} \log\left(2\pi e \cdot \sigma^2 (r^2 \cdot x_{\text{in}})\right)$$

The Bottleneck Entropy is the **minimum** combined entropy across all layers:

$$H_{\text{btn}} = \min_i \left(H_{\text{conv}}^{(i)} + H_{\text{img}}^{(i)}\right)$$

- **The Final MP Problem**
  The final optimization goal adds a penalty coefficient Q, while ω is a hyper-parameter that controls the size of Q.

$$\begin{aligned} &\max \sum_{s=3}^{M} \mathrm{PRscore}_i(\cdot) - \omega Q, \\ &\text{s.t. } H_i \leq \text{ budget}, \\ &\quad \mathrm{FLOPs}\,[f(\cdot)] \leq \text{ budget}, \\ &\quad Q \triangleq \exp\left[\mathrm{Var}\left(\frac{L_1}{\gamma_1}, \frac{L_2}{\gamma_2} \cdots \frac{L_M}{\gamma_M}\right)\right], \end{aligned} \tag{3}$$

References

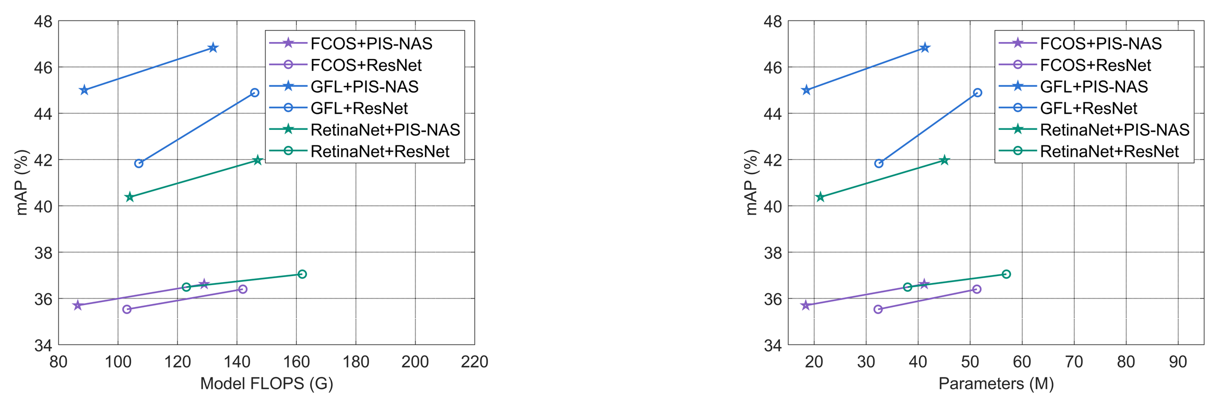
[1] Hervé Abdi. The kendall rank correlation coefficient. *Encyclopedia of measurement and statistics*, 2:508–510, 2007.

[2] Wenjie Luo, Yujia Li, Raquel Urtasun, and Richard Zemel. Understanding the effective receptive field in deep convolutional neural networks. *Advances in neural information processing systems*, 29, 2016.

[3] Xuan Shen, Yaohua Wang, Ming Lin, Yilun Huang, Hao Tang, Xiuyu Sun, and Yanzhi Wang. Deepmad: Mathematical architecture design for deep convolutional neural network. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR)*, 2023.

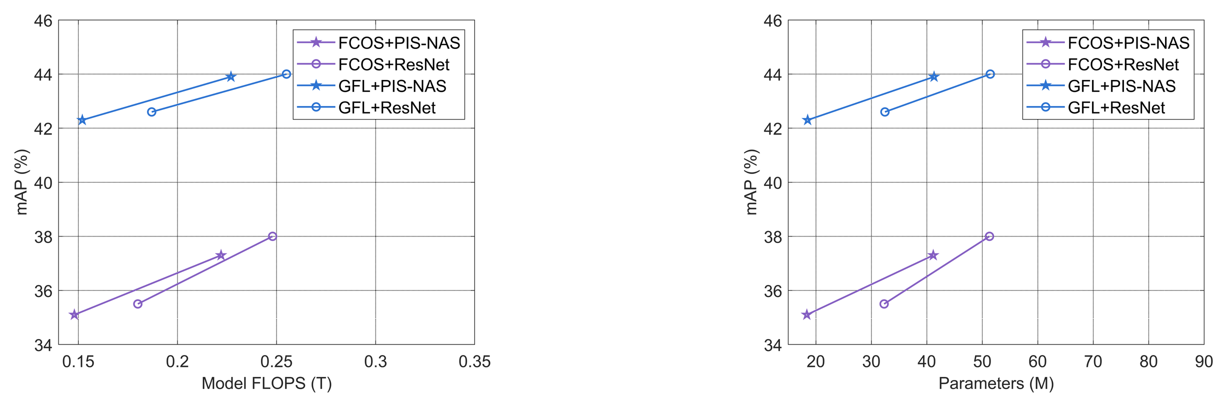
[4] Zhenhong Sun, Ming Lin, Xiuyu Sun, Zhiyu Tan, Hao Li, and Rong Jin. Mae-det: Revisiting maximum entropy principle in zero-shot nas for efficient object detection. In *International Conference on Machine Learning*. PMLR, 2022.

Experiments



(a) mAP vs. FLOPs (b) mAP vs. Parameters

Figure 2. mAP vs. FLOPs and mAP vs. Parameters on PASCAL VOC.



(a) mAP vs. FLOPs (b) mAP vs. Parameters

Figure 3. mAP vs. FLOPs and mAP vs. Parameters on MS COCO in Table.

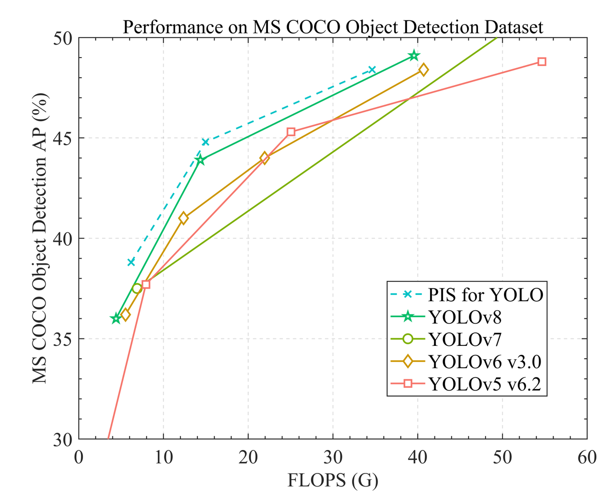


Figure 4. Comparison of YOLO object detectors on MS COCO.

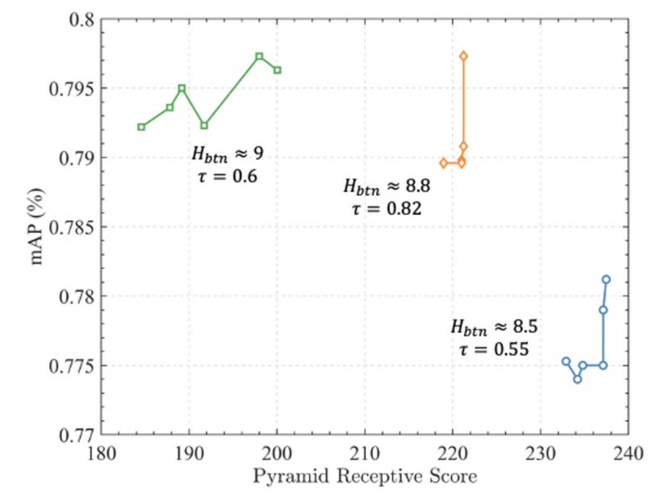


Figure 5. $H_{btn} = [8.5, 8.8, 9.0]$. Kendall coefficient τ [1] is used to measure the correlation.

Table 1. Comparisons with SOTA NAS methods for object detection

Method	Venue	Training-free	Search Cost GPU hours	Serach Part	FLOPs All	Pretrain/ Scratch	Epochs	COCO AP_{test}
DetNAS	NeurIPS'19	×	1632	backbone	289G	Pretrain	24	43.4
SP-NAS	CVPR'20	×	624	backbone	655G	Pretrain	24	47.4
SpineNet	CVPR'20	×	100x TPUv3	backbone+FPN	524G	Scratch	350	48.1
MAE-DET	ICML'22	✓	15	backbone	279G	Scratch	73	48.0
DeepMAD	CVPR'23	✓	0.1	backbone	197G	Pretrain	24	44.9
PIS-NAS	-	✓	8	backbone	**88G**	Scratch	450	**48.8**