Zen-NAS: A Zero-Shot NAS for High-Performance Image Recognition

Department of Computer Engineering

Dongseo University

Jie Yong Shin,

2023.03.16

Zen-NAS: A Zero-Shot NAS for High-Performance Image Recognition

Ming Lin
Alibaba Group
Bellevue, Washington, USA
ming.l@alibaba-inc.com

Pichao Wang
Alibaba Group
Bellevue, Washington, USA
pichao.wang@alibaba-inc.com

Zhenhong Sun
Alibaba Group
Hangzhou, Zhejiang, China
zhenhong.szh@alibaba-inc.com

Hesen Chen
Alibaba Group
Hangzhou, Zhejiang, China
hesen.chs@alibaba-inc.com

Xiuyu Sun
Alibaba Group
Hangzhou, Zhejiang, China
xiuyu.sxy@alibaba-inc.com

Qi Qian
Alibaba Group
Bellevue, Washington, USA
qi.qian@alibaba-inc.com

Hao Li
Alibaba Group
Hangzhou, Zhejiang, China
lihao.lh@alibaba-inc.com

Rong Jin
Alibaba Group
Hangzhou, Zhejiang, China
jinrong.jr@alibaba-inc.com

- Published at ICCV on 2021
- From Alibaba Group

CONTENTS

1. Highlights

1. Why did they develop this?
   - Gaussian complexity(pi-score) has numerical overflow problem to be used as a metric for a network's expressivity.

2. What are contributions of this paper?
   - They have shown a novel metric to measure a neural network architecture.
   - They have shown that this metric is faster than previous metric and can achieve higher accuracy compared to other NAS methods
   - They have achieved 83.5% of accuracy with NAS on ImageNet dataset within half of a GPU day.

3. What are possible improvements?

Difference between pi-Score and Zen-Score

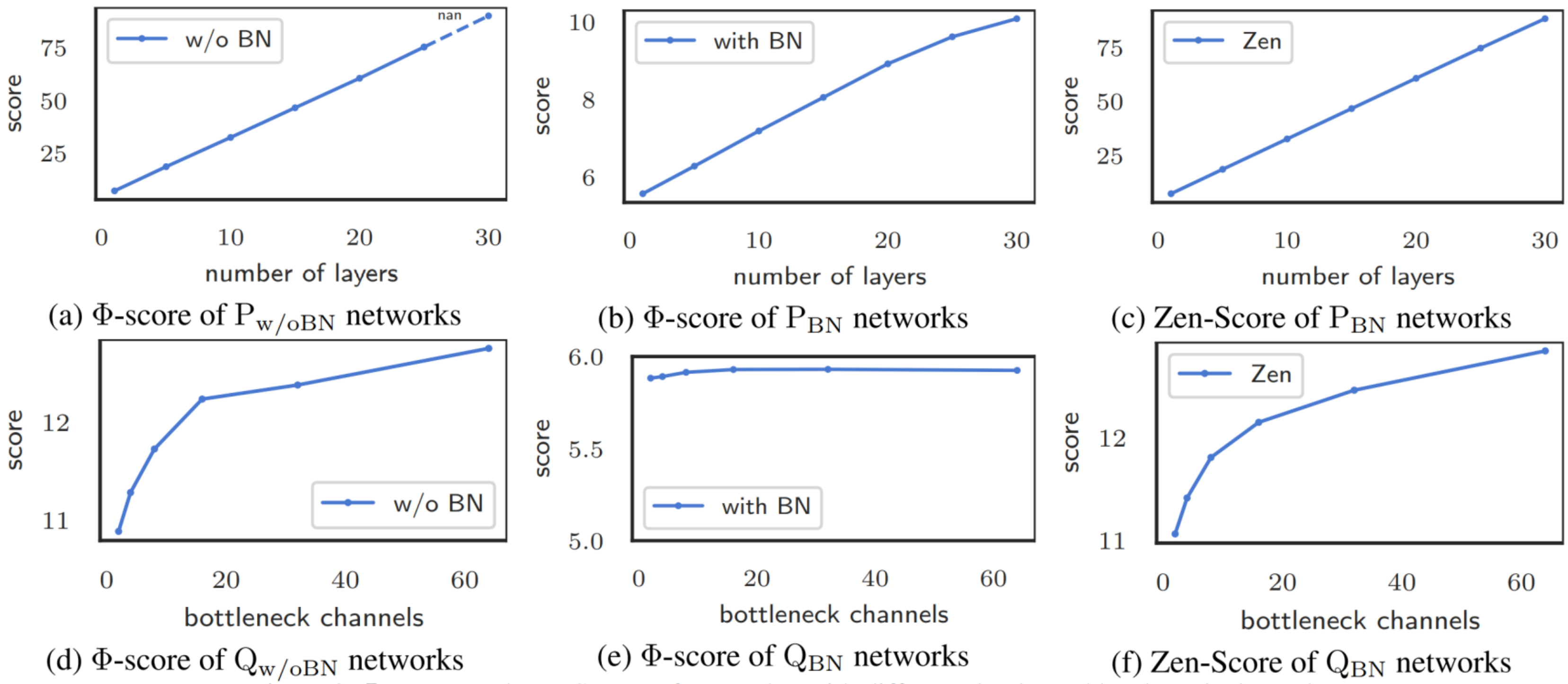

Figure 2. Φ-scores and Zen-Scores of networks, with different depths and bottleneck channels.

- When it comes to bottleneck channels, previous metric score (Gaussian complexity) incurs numerical overflow

2. Zen-Score

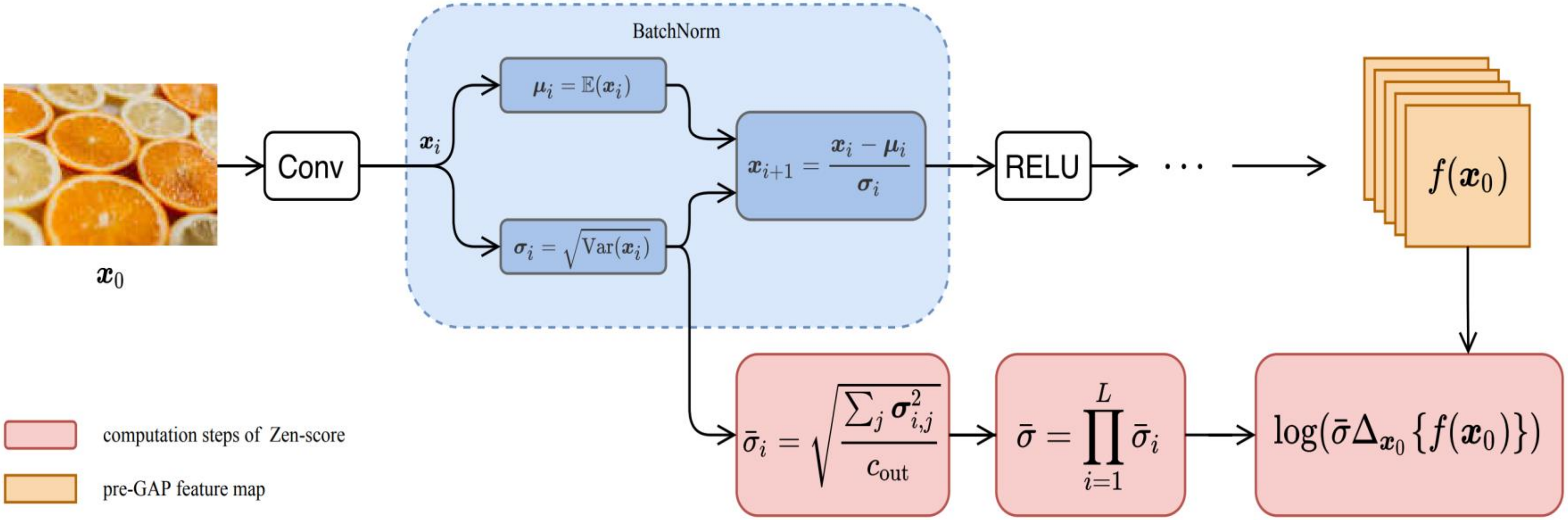

Figure 3. Zen-Score computational graph. x_0 is one mini-batch of input images. For each BN layer, we extract its mini-batch deviation parameter σ_i. $\Delta_{x_0}\{f(x_0)\}$ is the differential of pre-GAP feature map $f(x_0)$ with respect to x_0.

2. Zen-Score

Algorithm 1 Zen-Score

Require: Network $\mathcal{F}(\cdot)$ with pre-GAP feature map $f(\cdot)$; $\alpha = 0.01$.

Ensure: Zen-Score $\mathrm{Zen}(\mathcal{F})$.

1: Remove all residual links in $\mathcal{F}$.

2: Initialize all neurons in $\mathcal{F}$ by $\mathcal{N}(0, 1)$.

3: Sample $\boldsymbol{x}, \boldsymbol{\epsilon} \sim \mathcal{N}(0, 1)$.

4: Compute $\Delta \triangleq \mathbb{E}_{\boldsymbol{x},\boldsymbol{\epsilon}} \| f(\boldsymbol{x}) - f(\boldsymbol{x} + \alpha\boldsymbol{\epsilon}) \|_F$.

5: For the i-th BN layer with m output channels, compute $\bar{\sigma}_i = \sqrt{\sum_j \boldsymbol{\sigma}_{i,j}^2 / m}$ where $\boldsymbol{\sigma}_{i,j}$ is the mini-batch standard deviation statistic of the j-th channel in BN.

6: $\mathrm{Zen}(F) \triangleq \log(\Delta) + \sum_i \log(\bar{\sigma}_i)$.

- All residual links are removed as pre-processing
- Sample input vector and perturb with Gaussian noise
- Δ : Perturbation (replaces gradient of x)
- The Zen-Score of network with BN layers approximates the Φ-score of the same network without BN layers

3. Zen-NAS

Algorithm 2 Zen-NAS

Require: Search space $\mathcal{S}$, inference budget B, maximal depth L, total number of iterations T, evolutionary population size N, initial structure F_0.

Ensure: NAS-designed ZenNet F^*.

```
Initialize population P = {F_0}.
for t = 1, 2, ··· , T do
    Randomly select F_t ∈ P.
    Mutate F̂_t = MUTATE(F_t, S)
    if F̂_t exceeds inference budget or has more than L layers then
        Do nothing.
    else
        Get Zen-Score z = Zen(F̂_t).
        Append F̂_t to P.
    end if
    Remove network of the smallest Zen-Score if the size of P exceeds B.
end for
Return F*, the network of the highest Zen-Score in P.
```

Algorithm 3 MUTATE

Require: Structure F_t, search space $\mathcal{S}$.

Ensure: Randomly mutated structure $\hat{F}_t$.

```
Uniformly select a block h in F_t.
Uniformly alternate the block type, kernel size, width and depth of h within some range.
Return the mutated structure F̂_t.
```

- Inference cost < Budget
- Maximal depth < L
- Remove lowest Zen-Score individuals

Neural Architecture Search on ImageNet

NAS	Method	Top-1 (%)	GPU Day
AmoebaNet-A [41]	EA	74.5	3150†
EcoNAS [69]	EA	74.8	8
CARS-I [62]	EA	75.2	0.4
GeNet [57]	EA	72.1	17
DARTS [26]	GD	73.1	4
SNAS [59]	GD	72.7	1.5
PC-DARTS [61]	GD	75.8	3.8
ProxylessNAS [6]	GD	75.1	8.3
GDAS [66]	GD	74	0.8
FBNetV2-L1 [54]	GD	77.2	25
NASNet-A [70]	RL	74	1800
Mnasnet-A [49]	RL	75.2	-
MetaQNN [3]	RL	77.4	96
PNAS [25]	SMBO	74.2	224
SemiNAS [29]	SSL	76.5	4
TE-NAS [7]	ZS	74.1	0.2
OFANet [5]	PS	80.1	51.6
EfficientNet-B7 [50]	Scaling	84.4	3800‡
Zen-NAS	ZS	83.6	0.5

5. Conclusion and Further research

1. They have a new metric (Zen-Score) from previous studies (Zero-Shot proxy).

2. By using Zen-Score, they have achieved State-Of-The-Art (SOTA) in ImageNet dataset.

3. According to them, RL based method can be implemented with Zen-Score.

4. In the algorithm, they do not keep the elite. By keeping the elite, it can have higher accuracy.

References

- Zen-NAS: A Zero-Shot NAS for High-Performance Deep Image Recognition https://paperswithcode.com/paper/zen-nas-a-zero-shot-nas-for-high-performance

- Zero-Cost Proxies for Lightweight NAS https://arxiv.org/abs/2101.08134

- EfficientNet: Rethinking Model Scaling for Convolutional Neural Networks https://arxiv.org/abs/1905.11946

- Neural Architecture Search on ImageNet in Four GPU Hours: A Theoretically Inspired Perspective (TE-NAS) https://arxiv.org/abs/2102.11535

Thank you

Q & A

Zen-Score with other metrics

proxy	CIFAR-10	CIFAR-100
Zen-Score	**96.2%**	**80.1%**
FLOPs	93.1%	64.7%
grad	92.8%	65.4%
synflow	95.1%	75.9%
TE-Score	96.1%	77.2%
NASWOT	96.0%	77.5%
Random	93.5±0.7%	71.1±3.1%

Table 1. Top-1 accuracies on CIFAR-10/CIFAR-100 for five zero-shot proxies. Budget: model size $N \leq 1\,\mathrm{M}$. 'Random': average accuracy $\pm$ std for random search.

2. Zen-Score

Theorem 1. *Let $\bar{f}(\boldsymbol{x}_0) = \bar{\boldsymbol{x}}_L$ be an L-layer vanilla network without BN layers. $f(\boldsymbol{x}_0) = \boldsymbol{x}_L$ is its sister network with BN layers. For some constants $0 < \delta < 1$, $K_0 \leq \mathcal{O}[\sqrt{\log(1/\delta)}]$, when $BHW \geq \mathcal{O}[(LK_0)^2]$ is large enough, with probability at least $1-\delta$, we have*

$$(1 - L\epsilon)^2 \leq \frac{(\prod_{t=1}^{L} \bar{\sigma}_t^2)\mathbb{E}_\theta\{\|\boldsymbol{x}_L\|^2\}}{\mathbb{E}_\theta\|\bar{\boldsymbol{x}}_L\|^2} \leq (1 + L\epsilon)^2 \quad (4)$$

where $\epsilon \triangleq \mathcal{O}(2K_0/\sqrt{BHW})$.

Informally speaking, Theorem 1 says that to compute $\|\bar{f}(\cdot)\|$, we only need to compute $\|f(\cdot)\|$ then re-scale with $\prod_{t=1}^{L} \bar{\sigma}_t$. The approximation error is bounded by $L\epsilon$. By taking gradient of $\boldsymbol{x}$ on both $\bar{f}(\cdot)$ and $f(\cdot)$, we obtain the desired relationship between Zen-Score and Φ-score.

- Zen-Score of a network approximates pi-score

Zen-Score with other Zero-Shot Proxies

proxy	model	N	time	speed-up
TE-Score	ResNet-18	16	0.34	1/28x
	ResNet-50	16	0.77	1/20x
NASWOT†	ResNet-18	16	0.040	1/3.3x
	ResNet-50	16	0.059	1/1.6x
Zen-Score	ResNet-18	16	0.012	1.0
	ResNet-50	16	0.037	1.0

Table 2. Time cost (in seconds) of computing Zen/TE-Score for ResNet-18/50 at resolution 224x224. The statistical error is within 5%. 'time': time for computing Zen/TE-score for N images, measured in seconds, averaged over 100 trials. 'speed-up': speed-up rate of TE-Score v.s. Zen-Score.
†: The official implementation outputs Inf score for ResNet-18/50.

Comparison with the limit of FLOPs

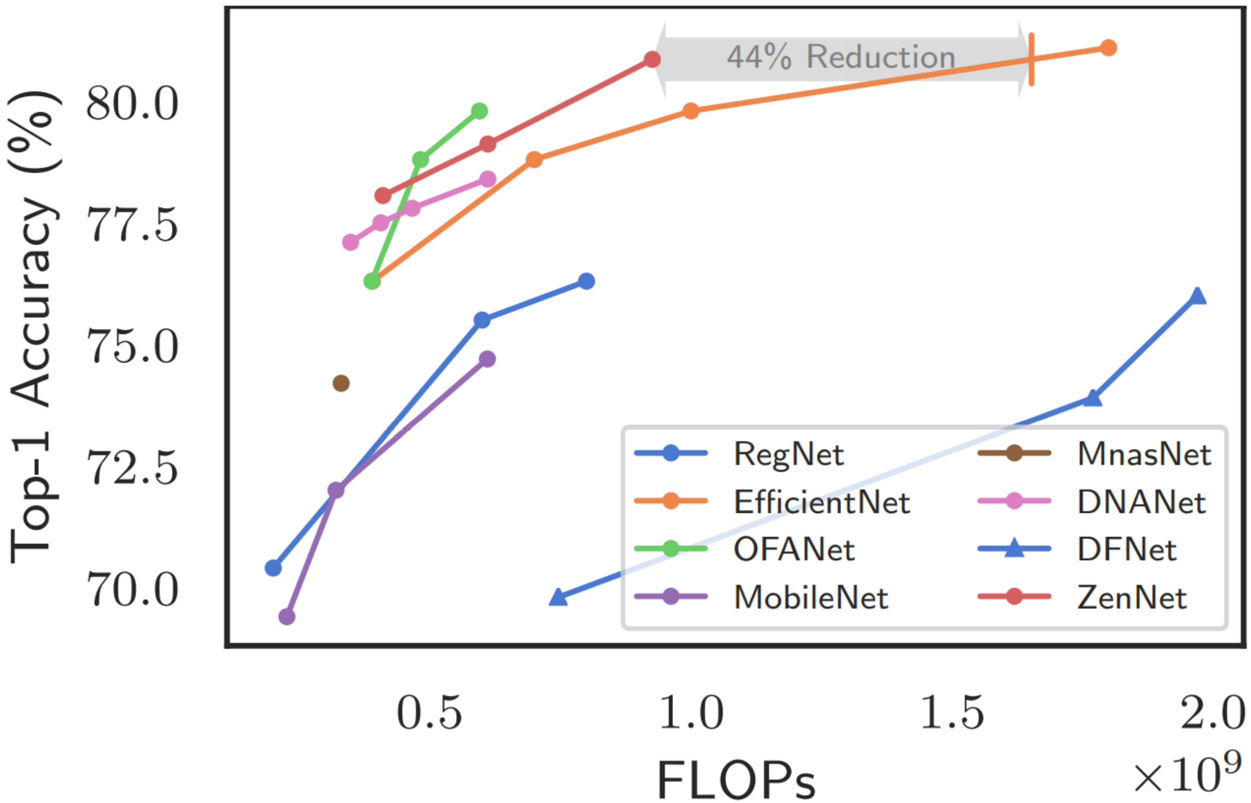

Figure 4. ZenNets optimized for FLOPs.

Comparison with other NAS algorithms

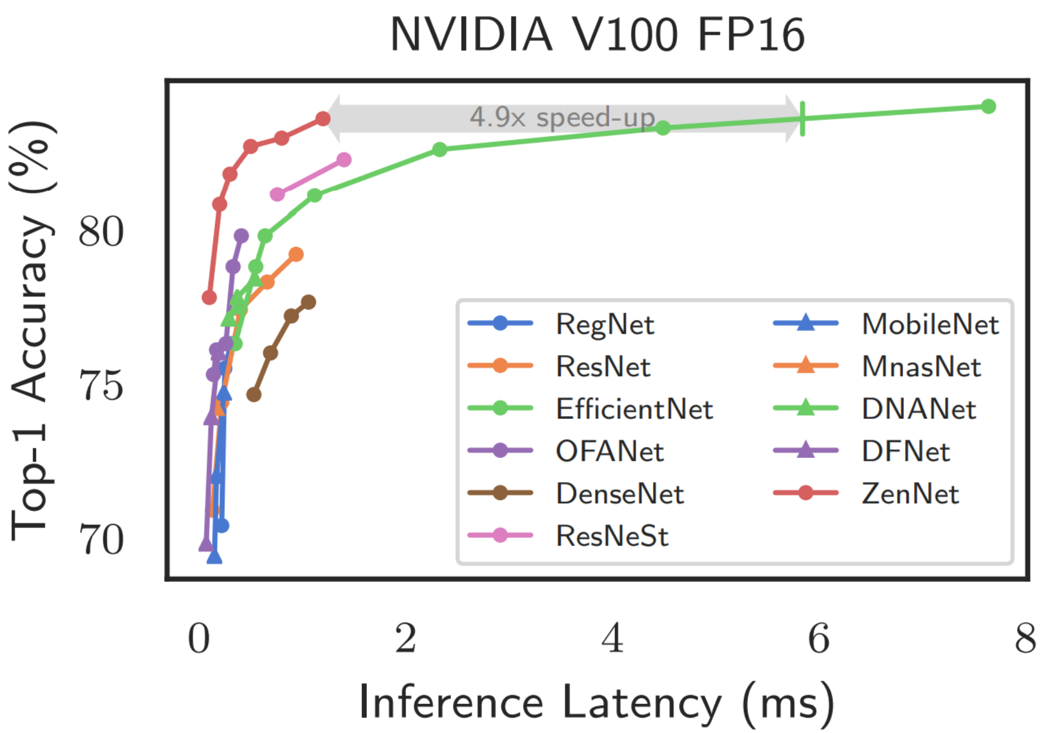

Figure 1. ZenNets top-1 accuracy v.s. inference latency (milliseconds per image) on ImageNet. Benchmarked on NVIDIA V100 GPU, half precision (FP16), batch size 64, searching cost 0.5 GPU day.