On Data Augmentation Trends in Deep Learning (especially low-level vision)

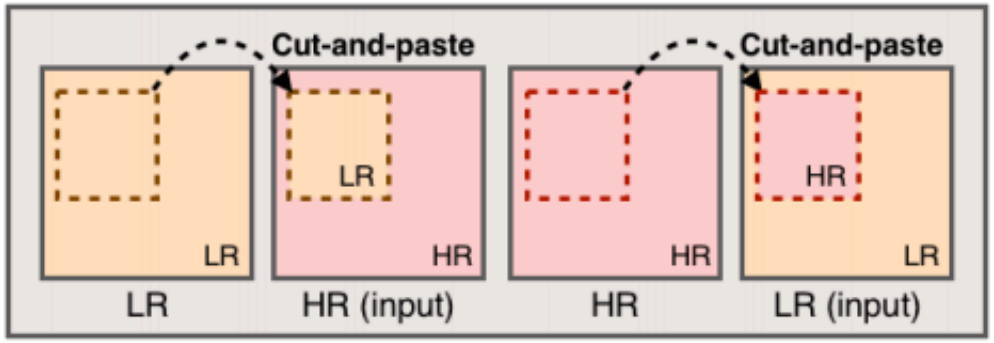
Seobin Park

Basic Image Augmentation in low-level vision

- Cropping, flipping, rotating are the most basic data augmentation method.

Newer augmentation methods

(a) High resolution (b) Low resolution (c) CutBlur

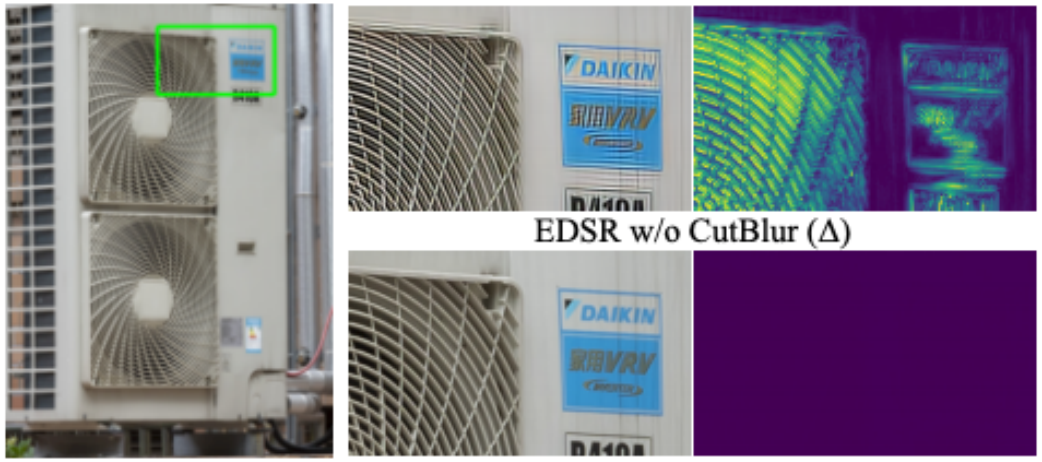


(d) Schematic illustration of CutBlur operation

(e) Blend (f) RGB permute (g) Cutout (25%) [8] (h) Mixup [37] (i) CutMix [36] (j) CutMixup

Reference: https://arxiv.org/pdf/2004.00448.pdf

CutBlur Results



HR (input)

EDSR w/o CutBlur (Δ)

EDSR w/ CutBlur (Δ)

Table 1. PSNR (dB) comparison of different data augmentation methods in super-resolution. We report the baseline model (EDSR [18]) performance that is trained on DIV2K (×4) [2] and RealSR (×4) [5]. The models are trained from scratch. δ denotes the performance gap between with and without augmentation.

Method	DIV2K (δ)	RealSR (δ)
EDSR	29.21 (+0.00)	28.89 (+0.00)
Cutout [8] (0.1%)	29.22 (+0.01)	28.95 (+0.06)
CutMix [36]	29.22 (+0.01)	28.89 (+0.00)
Mixup [37]	29.26 (+0.05)	28.98 (+0.09)
CutMixup	29.27 (+0.06)	29.03 (+0.14)
RGB perm.	29.30 (+0.09)	29.02 (+0.13)
Blend	29.23 (+0.02)	29.03 (+0.14)
CutBlur	**29.26 (+0.05)**	**29.12 (+0.23)**
All DA's (random)	**29.30 (+0.09)**	**29.16 (+0.27)**

Reference: https://arxiv.org/pdf/2004.00448.pdf

Automatic Data Augmentation

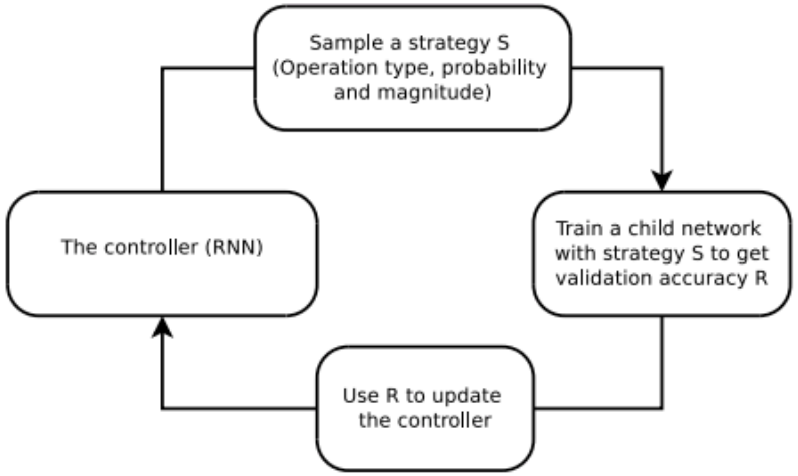


Figure 1. Overview of our framework of using a search method (e.g., Reinforcement Learning) to search for better data augmentation policies. A controller RNN predicts an augmentation policy from the search space. A child network with a fixed architecture is trained to convergence achieving accuracy R. The reward R will be used with the policy gradient method to update the controller so that it can generate better policies over time.

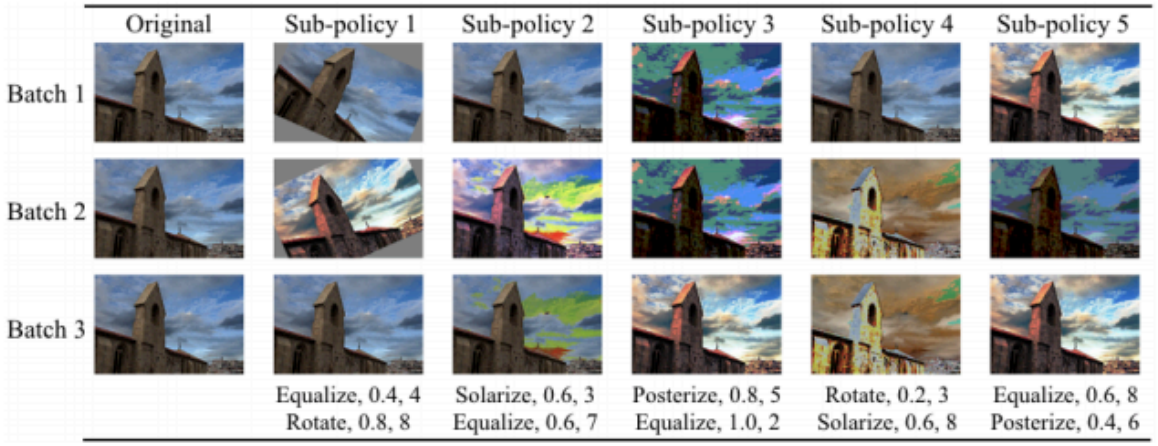


Figure 3. One of the successful policies on ImageNet. As described in the text, most of the policies found on ImageNet used color-based transformations.

Reference: https://arxiv.org/pdf/1805.09501.pdf

Differentiable Automatic Data Augmentation (DADA)

$$\min \ \mathcal{L}_{\text{val}}(w^*(d)),$$
$$\text{s.t.} \quad w^*(d) = \text{argmin}_w \ \mathbb{E}[\mathcal{L}_{\text{train}}(w, d)].$$

$$\mathcal{L}_{\text{val}}(w_k - \zeta \nabla_w \mathbb{E}[\mathcal{L}_{\text{train}}(w_k, d_{k-1})]).$$

(a) GPU hours

Dataset	AA [3]	PBA [10]	Fast AA [15]	OHL AA [16]	**DADA**
CIFAR-10	5000	5	3.5	83.4*	0.1
CIFAR-100	-	-	-	-	0.2
SVHN	1000	1	1.5	-	0.1
ImageNet	15000	-	450	625*	1.3

(b) Test set error rate (%)

Dataset	AA [3]	PBA [10]	Fast AA [15]	OHL AA [16]	**DADA**
CIFAR-10	2.6	2.6	2.7	2.6	2.7
CIFAR-100	17.1	16.7	17.3	-	17.5
SVHN	1.1	1.2	1.1	-	1.2
ImageNet	22.4	-	22.4	21.1	22.5

Table 2: CIFAR-10 and CIFAR-100 test error rates (%).

Dataset	Model	Baseline	Cutout [4]	AA [3]	PBA [10]	Fast AA [15]	**DADA**
CIFAR-10	Wide-ResNet-40-2	5.3	4.1	3.7	-	3.6	3.6
CIFAR-10	Wide-ResNet-28-10	3.9	3.1	2.6	2.6	2.7	2.7
CIFAR-10	Shake-Shake(26 2x32d)	3.6	3.0	2.5	2.5	2.7	2.7
CIFAR-10	Shake-Shake(26 2x96d)	2.9	2.6	2.0	2.0	2.0	2.0
CIFAR-10	Shake-Shake(26 2x112d)	2.8	2.6	1.9	2.0	2.0	2.0
CIFAR-10	PyramidNet+ShakeDrop	2.7	2.3	1.5	1.5	1.8	1.7
CIFAR-100	Wide-ResNet-40-2	26.0	25.2	20.7	-	20.7	20.9
CIFAR-100	Wide-ResNet-28-10	18.8	18.4	17.1	16.7	17.3	17.5
CIFAR-100	Shake-Shake(26 2x96d)	17.1	16.0	14.3	15.3	14.9	15.3
CIFAR-100	PyramidNet+ShakeDrop	14.0	12.2	10.7	10.9	11.9	11.2

Reference: https://www.ecva.net/papers/eccv_2020/papers_ECCV/papers/123670579.pdf