Implicit Semantic Data Augmentation for Deep Networks

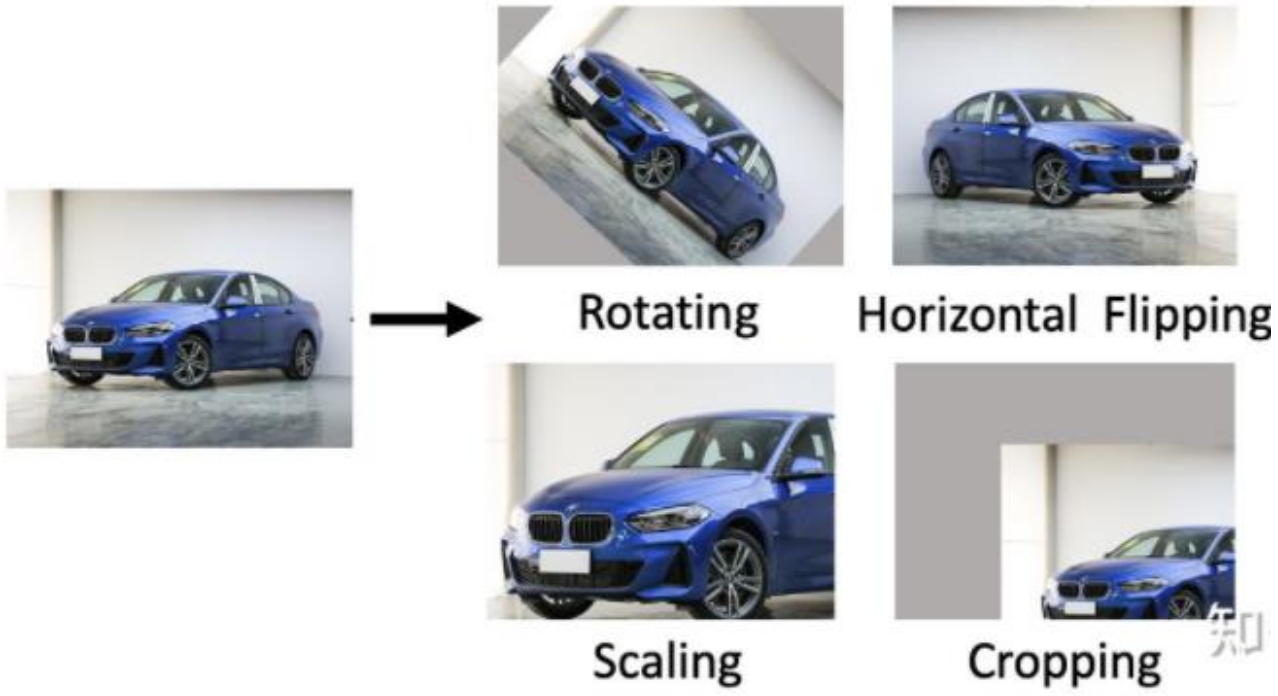
Yulin Wang[1*] **Xuran Pan**[1*] **Shiji Song**[1] **Hong Zhang**[2] **Cheng Wu**[1] **Gao Huang**[1†]
[1]Department of Automation, Tsinghua University, Beijing, China
Beijing National Research Center for Information Science and Technology (BNRist),
[2]Baidu Inc., China
{yulin.bh, fykalviny}@gmail.com, pxr18@mails.tsinghua.edu.cn,
{shijis, wuc, gaohuang}@tsinghua.edu.cn

NIPS 2019

Data Augmentation

Data Augmentation is a widely used technique to alleviate
overfitting in training deep networks in the case of a small number of data sets.

☐ Traditional Data Augmentation



Translation

Data Augmentation

Generative adversarial network

- Complicated
- High cost of time and computation
- Relying on large amounts of data

ISDA

Implicit Semantic Data Augmentation

This work is inspired by the observation that certain directions in the feature space correspond to meaningful semantic transformations.

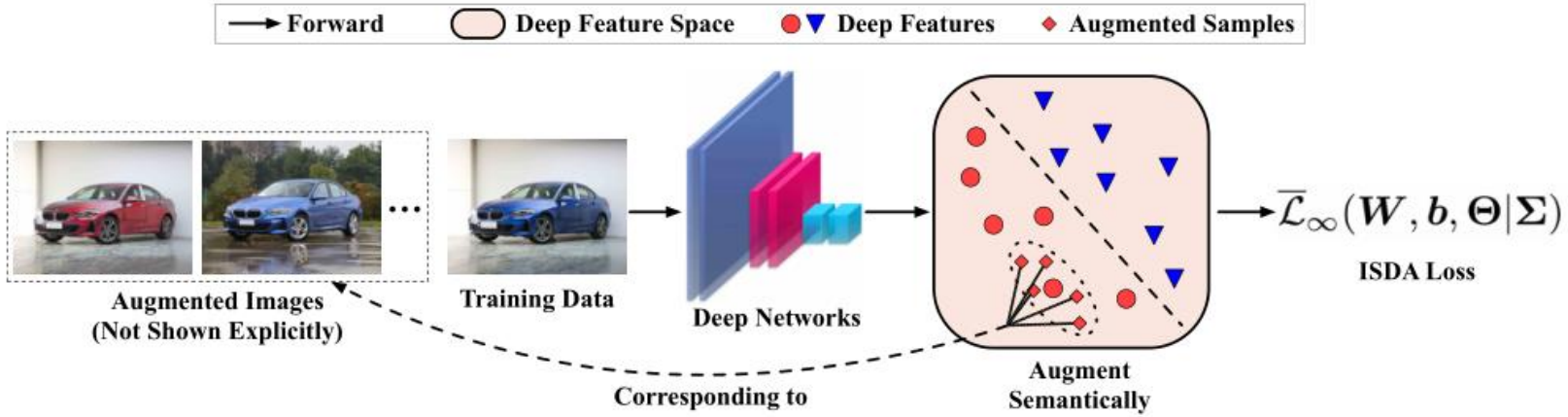
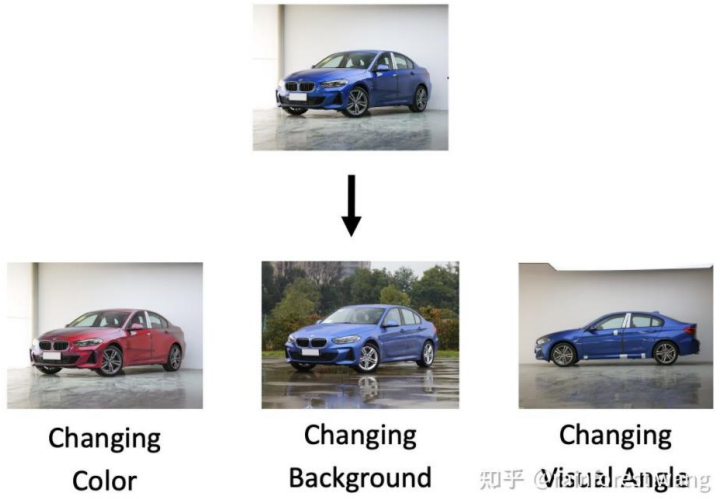
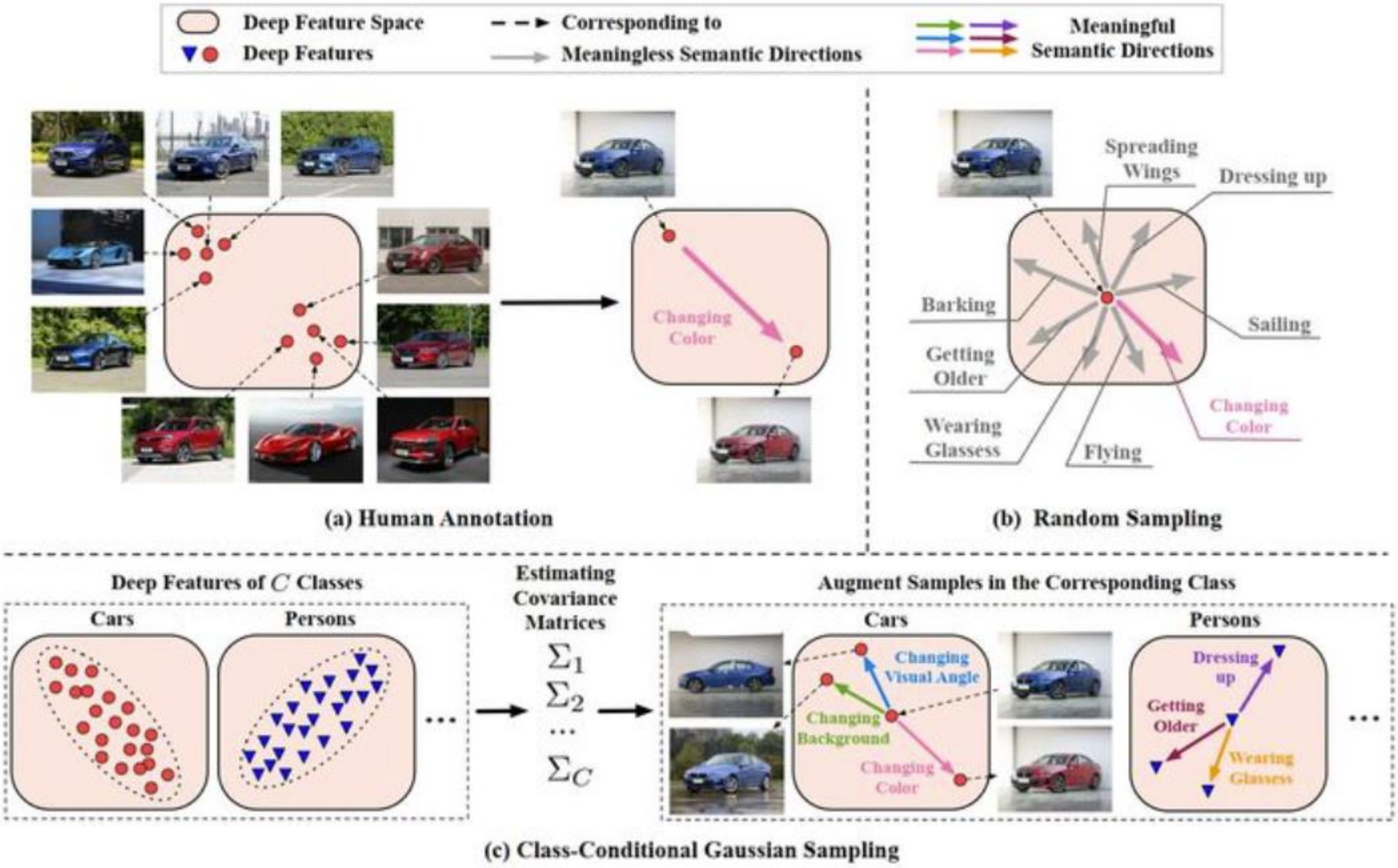


Figure 1: An overview of ISDA. Inspired by the observation that certain directions in the feature space correspond to meaningful semantic transformations, we augment the training data semantically by translating their features along these semantic directions, without involving auxiliary deep networks. The directions are obtained by sampling random vectors from a zero-mean normal distribution with dynamically estimated class-conditional covariance matrices. In addition, instead of performing augmentation explicitly, ISDA boils down to minimizing a closed-form upper-bound of the expected cross-entropy loss on the augmented training set, which makes our method highly efficient.



ISDA

How to find meaningful semantic transformations?



ISDA

× Manually searching for semantic directions is infeasible for large scale problems.

☐ Online estimation of class-conditional covariance matrices.

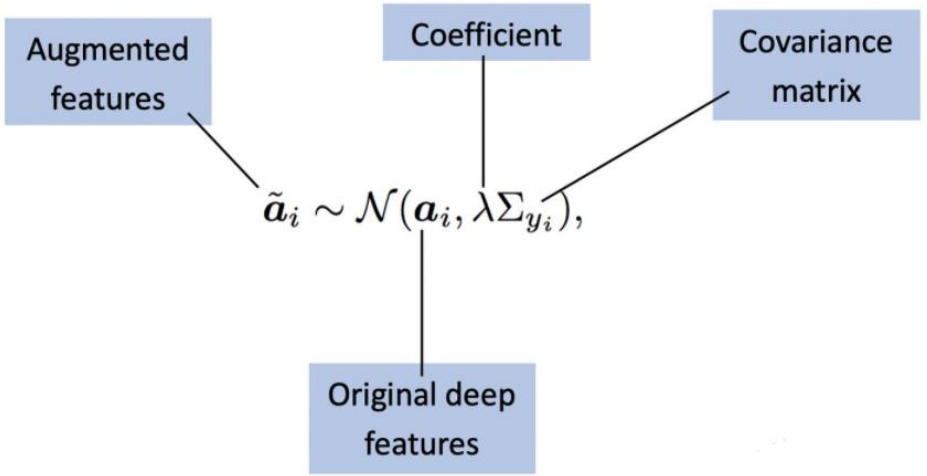
Then randomly sample vectors from a zero-mean multivariate normal distribution.

$$\mathcal{N}(0, \Sigma_{y_i})$$

covariance matrices for every class

$$\boldsymbol{a}_i = [a_{i1}, \ldots, a_{iA}]^T = G(\boldsymbol{x}_i, \boldsymbol{\Theta})$$

$$\tilde{\boldsymbol{a}}_i \sim \mathcal{N}(\boldsymbol{a}_i, \lambda \Sigma_{y_i}) \quad \lambda = (t/T) \times \lambda_0$$



Dynamic estimation of covariance matrices

1. Estimation of average values of the features of j class at t step.

2. Estimation of average values of the features of j class s in t mini-batch.

$$\boldsymbol{\mu}_j^{(t)} = \frac{n_j^{(t-1)}\boldsymbol{\mu}_j^{(t-1)} + m_j^{(t)}\boldsymbol{\mu'}_j^{(t)}}{n_j^{(t-1)} + m_j^{(t)}}, \tag{9}$$

$$\Sigma_j^{(t)} = \frac{n_j^{(t-1)}\Sigma_j^{(t-1)} + m_j^{(t)}\Sigma'^{(t)}_j}{n_j^{(t-1)} + m_j^{(t)}} + \frac{n_j^{(t-1)} m_j^{(t)}(\boldsymbol{\mu}_j^{(t-1)} - \boldsymbol{\mu'}_j^{(t)})(\boldsymbol{\mu}_j^{(t-1)} - \boldsymbol{\mu'}_j^{(t)})^T}{(n_j^{(t-1)} + m_j^{(t)})^2}, \tag{10}$$

$$n_j^{(t)} = n_j^{(t-1)} + m_j^{(t)} \tag{11}$$

3.Total number of training samples belonging to j class in all t mini-batches

4.Total number of training samples belonging to j class in t mini-batches

Novel Loss

$$\{(\boldsymbol{a}_i^1, y_i), \ldots, (\boldsymbol{a}_i^M, y_i)\}_{i=1}^N$$

cross-entropy (CE) loss:

$$\mathcal{L}_M(\boldsymbol{W}, \boldsymbol{b}, \boldsymbol{\Theta}) = \frac{1}{N}\sum_{i=1}^{N}\frac{1}{M}\sum_{k=1}^{M} -log(\frac{e^{\boldsymbol{w}_{y_i}^T \boldsymbol{a}_i^k + b_{y_i}}}{\sum_{j=1}^{C} e^{\boldsymbol{w}_j^T \boldsymbol{a}_i^k + b_j}}), \tag{2}$$

M -> ∞

Final goal

$$\mathcal{L}_\infty(\boldsymbol{W}, \boldsymbol{b}, \boldsymbol{\Theta}|\boldsymbol{\Sigma}) \leq \frac{1}{N}\sum_{i=1}^{N} -log(\frac{e^{\boldsymbol{w}_{y_i}^T \boldsymbol{a}_i + b_{y_i}}}{\sum_{j=1}^{C} e^{\boldsymbol{w}_j^T \boldsymbol{a}_i + b_j + \frac{\lambda}{2}(\boldsymbol{w}_j^T - \boldsymbol{w}_{y_i}^T)\Sigma_{y_i}(\boldsymbol{w}_j - \boldsymbol{w}_{y_i})}}) \triangleq \overline{\mathcal{L}}_\infty. \tag{4}$$

ISDA

Proof.

$$\mathcal{L}_\infty(\boldsymbol{W}, \boldsymbol{b}, \boldsymbol{\Theta}|\boldsymbol{\Sigma}) = \frac{1}{N}\sum_{i=1}^{N} \mathrm{E}_{\tilde{\boldsymbol{a}}_i}[log(\sum_{j=1}^{C} e^{(\boldsymbol{w}_j^T - \boldsymbol{w}_{y_i}^T)\tilde{\boldsymbol{a}}_i + (b_j - b_{y_i})})] \tag{5}$$

Jensen's inequality

$$\mathrm{E}[logX] \leq log\mathrm{E}[X]$$

$$\leq \frac{1}{N}\sum_{i=1}^{N} log(\sum_{j=1}^{C} \mathrm{E}_{\tilde{\boldsymbol{a}}_i}[e^{(\boldsymbol{w}_j^T - \boldsymbol{w}_{y_i}^T)\tilde{\boldsymbol{a}}_i + (b_j - b_{y_i})}]) \tag{6}$$

$$= \frac{1}{N}\sum_{i=1}^{N} log(\sum_{j=1}^{C} e^{(\boldsymbol{w}_j^T - \boldsymbol{w}_{y_i}^T)\boldsymbol{a}_i + (b_j - b_{y_i}) + \frac{\lambda}{2}(\boldsymbol{w}_j^T - \boldsymbol{w}_{y_i}^T)\Sigma_{y_i}(\boldsymbol{w}_j - \boldsymbol{w}_{y_i})}) \tag{7}$$

Moment-generating function

$$= \overline{\mathcal{L}}_\infty. \tag{8}$$

$$\mathrm{E}[e^{tX}] = e^{t\mu + \frac{1}{2}\sigma^2 t^2}, \quad X \sim \mathcal{N}(\mu, \sigma^2)$$

Algorithm

Algorithm 1 The ISDA Algorithm.

Input: $\mathcal{D}$, λ_0
Randomly initialize $\boldsymbol{W}$, $\boldsymbol{b}$ and $\boldsymbol{\Theta}$
for $t = 0$ **to** T **do**
    Sample a mini-batch $\{\boldsymbol{x}_i, y_i\}_{i=1}^{B}$ from $\mathcal{D}$
    Compute $\boldsymbol{a}_i = G(\boldsymbol{x}_i, \boldsymbol{\Theta})$
    Estimate the covariance matrices Σ_1, Σ_2, ..., Σ_C
    Compute $\overline{\mathcal{L}}_\infty$ according to Eq. (4)
    Update $\boldsymbol{W}$, $\boldsymbol{b}$, $\boldsymbol{\Theta}$ with SGD
end for
Output: $\boldsymbol{W}$, $\boldsymbol{b}$ and $\boldsymbol{\Theta}$

Experiments

Table 1: Evaluation of ISDA on CIFAR with different models. The average test error over the last 10 epochs is calculated in each experiment, and we report mean values and standard deviations in three independent experiments. The best results are **bold-faced**.

Method	Params	CIFAR-10	CIFAR-100
ResNet-32 [4]	0.5M	7.39 ± 0.10%	31.20 ± 0.41%
ResNet-32 + ISDA	0.5M	**7.09 ± 0.12%**	**30.27 ± 0.34%**
ResNet-110 [4]	1.7M	6.76 ± 0.34%	28.67 ± 0.44%
ResNet-110 + ISDA	1.7M	**6.33 ± 0.19%**	**27.57 ± 0.46%**
SE-ResNet-110 [33]	1.7M	6.14 ± 0.17%	27.30 ± 0.03%
SE-ResNet-110 + ISDA	1.7M	**5.96 ± 0.21%**	**26.63 ± 0.21%**
Wide-ResNet-16-8 [34]	11.0M	4.25 ± 0.18%	20.24 ± 0.27%
Wide-ResNet-16-8 + ISDA	11.0M	**4.04 ± 0.29%**	**19.91 ± 0.21%**
Wide-ResNet-28-10 [34]	36.5M	3.82 ± 0.15%	18.53 ± 0.07%
Wide-ResNet-28-10 + ISDA	36.5M	**3.58 ± 0.15%**	**17.98 ± 0.15%**
ResNeXt-29, 8x64d [35]	34.4M	3.86 ± 0.14%	18.16 ± 0.13%
ResNeXt-29, 8x64d + ISDA	34.4M	**3.67 ± 0.12%**	**17.43 ± 0.25%**
DenseNet-BC-100-12 [5]	0.8M	4.90 ± 0.08%	22.61 ± 0.10%
DenseNet-BC-100-12 + ISDA	0.8M	**4.54 ± 0.07%**	**22.10 ± 0.34%**
DenseNet-BC-190-40 [5]	25.6M	3.52%	17.74%
DenseNet-BC-190-40 + ISDA	25.6M	**3.24%**	**17.42%**

Experiments

Table 2: Evaluation of ISDA with state-of-the-art *non-semantic* augmentation techniques. 'AA' refers to AutoAugment [32]. We report mean values and standard deviations in three independent experiments. The best results are **bold-faced**.

Dataset	Networks	Cutout [31]	Cutout + ISDA	AA [32]	AA + ISDA
CIFAR-10	Wide-ResNet-28-10 [34]	2.99 ± 0.06%	**2.83 ± 0.04%**	2.65 ± 0.07%	**2.56 ± 0.01%**
	Shake-Shake (26, 2x32d) [36]	3.16 ± 0.09%	**2.93 ± 0.03%**	2.89 ± 0.09%	**2.68 ± 0.12%**
	Shake-Shake (26, 2x112d) [36]	2.36%	**2.25%**	2.01%	**1.82%**
CIFAR-100	Wide-ResNet-28-10 [34]	18.05 ± 0.25%	**17.02 ± 0.11%**	16.60 ± 0.40%	**15.62 ± 0.32%**
	Shake-Shake (26, 2x32d) [36]	18.92 ± 0.21%	**18.17 ± 0.08 %**	17.50 ± 0.19%	**17.21 ± 0.33%**
	Shake-Shake (26, 2x112d) [36]	17.34 ± 0.28%	**16.24 ± 0.20 %**	15.21 ± 0.20%	**13.87 ± 0.26%**

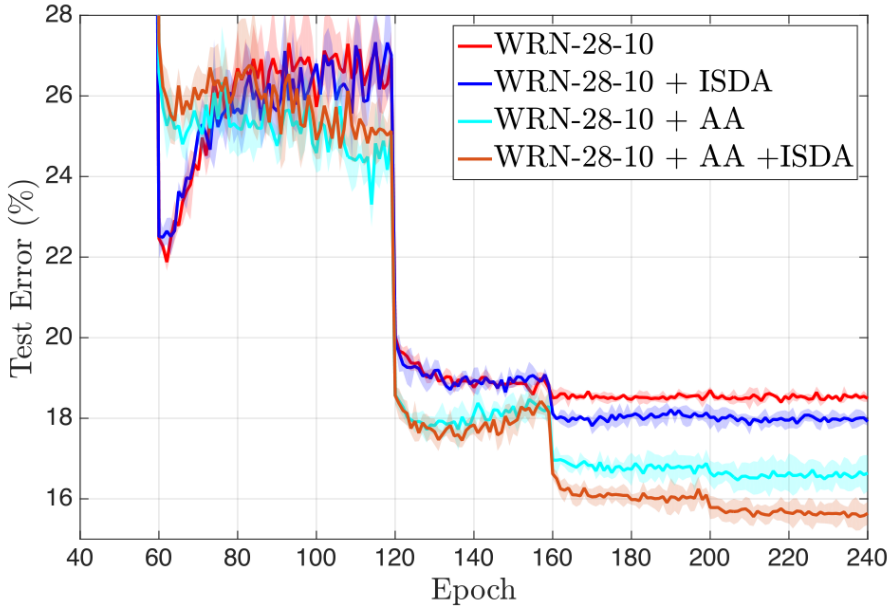


Figure 2: Curves of test errors on CIFAR-100 with Wide-ResNet (WRN).

Experiments

Table 3: Comparisons with the state-of-the-art methods. We report mean values and standard deviations of the test error in three independent experiments. Best results are **bold-faced**.

Method	ResNet-110		Wide-ResNet-28-10	
	CIFAR-10	CIFAR-100	CIFAR-10	CIFAR-100
Large Margin [18]	6.46±0.20%	28.00±0.09%	3.69±0.10%	18.48±0.05%
Disturb Label [38]	6.61±0.04%	28.46±0.32%	3.91±0.10%	18.56±0.22%
Focal Loss [17]	6.68±0.22%	28.28±0.32%	3.62±0.07%	18.22±0.08%
Center Loss [22]	6.38±0.20%	27.85±0.10%	3.76±0.05%	18.50±0.25%
L_q Loss [16]	6.69±0.07%	28.78±0.35%	3.78±0.08%	18.43±0.37%
WGAN [39]	6.63±0.23%	-	3.81±0.08%	-
CGAN [40]	6.56±0.14%	28.25±0.36%	3.84±0.07%	18.79±0.08%
ACGAN [41]	6.32±0.12%	28.48±0.44%	3.81±0.11%	18.54±0.05%
infoGAN [42]	6.59±0.12%	27.64±0.14%	3.81±0.05%	18.44±0.10%
Basic	6.76±0.34%	28.67±0.44%	-	-
Basic + Dropout	6.23±0.11%	27.11±0.06%	3.82±0.15%	18.53±0.07%
ISDA	6.33±0.19%	27.57±0.46%	-	-
ISDA + Dropout	**5.98±0.20%**	**26.35±0.30%**	**3.58±0.15%**	**17.98±0.15%**

For generator-based semantic augmentation methods

Wide-ResNet-28-10 and ResNeXt-29, 8x64d, our method outperforms the competitive baselines by nearly 0.7%. Compared to ResNets, DenseNets generally suffer less from overfitting due to their architecture design, thus appear to benefit less from our algorithm.

Experiments

Table 4: Evaluation of ISDA on ImageNet.

Method	Top-1	Top-5
ResNet-50 [4]	23.58%	6.92%
ResNet-50 + ISDA	**23.30%**	**6.82%**
ResNet-152 [4]	21.65%	6.01%
ResNet-152 + ISDA	**21.20%**	**5.67%**
ResNeXt-50, 32x4d [35]	22.42%	6.42%
ResNeXt-50, 32x4d + ISDA	**21.88%**	**6.23%**

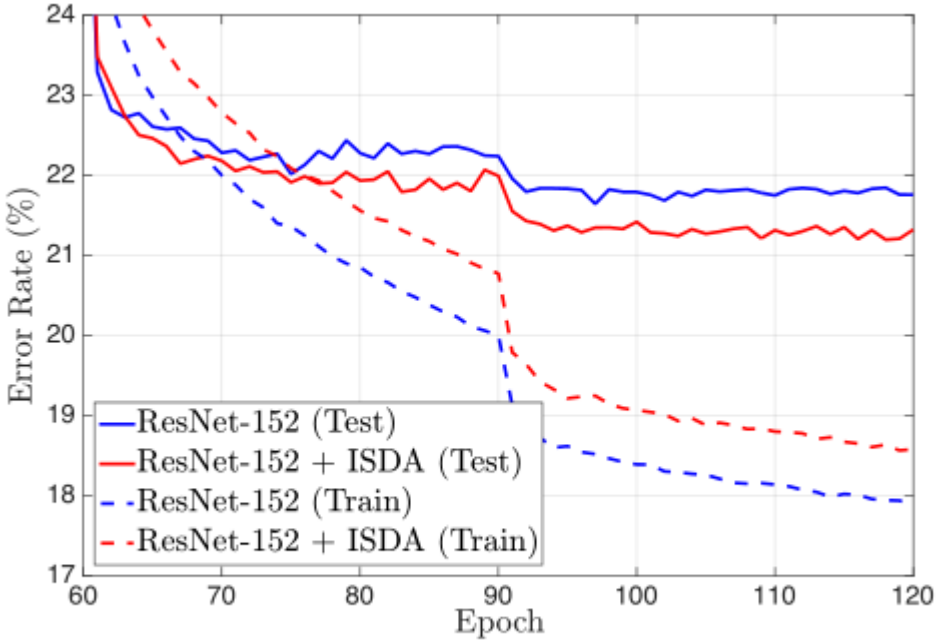


Figure 3: Training and test errors on ImageNet.

Experiments

Table 5: The ablation study for ISDA.

Setting	CIFAR-10	CIFAR-100
Basic	3.82±0.15%	18.58±0.10%
Identity matrix	3.63±0.12%	18.53±0.02%
Diagonal matrix	3.70±0.15%	18.23±0.02%
Single covariance matrix	3.67±0.07%	18.29±0.13%
Constant λ_0	3.69±0.08%	18.33±0.16%
ISDA	**3.58±0.15%**	**17.98±0.15%**

Experiments

Visualization Results

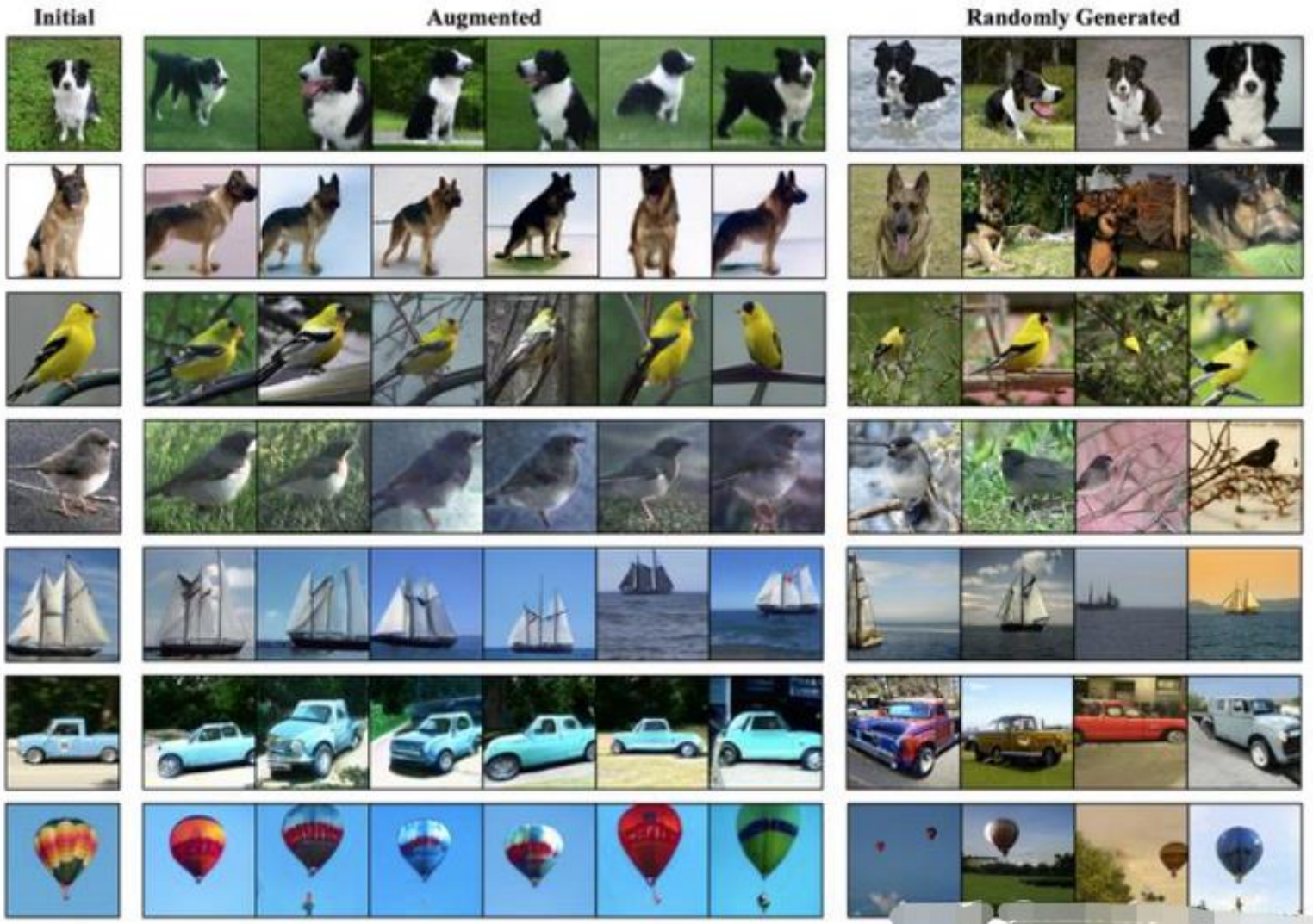


Fig. 7. Visualization of the semantically augmented images on ImageNet. ISDA is able to alter the semantics of images that are unrelated to the class identity, like backgrounds, actions of animals, visual angles, etc. We also present the randomly generated images of the same class.

Thanks