Mixup Augmentation

Outline

Data augmentation

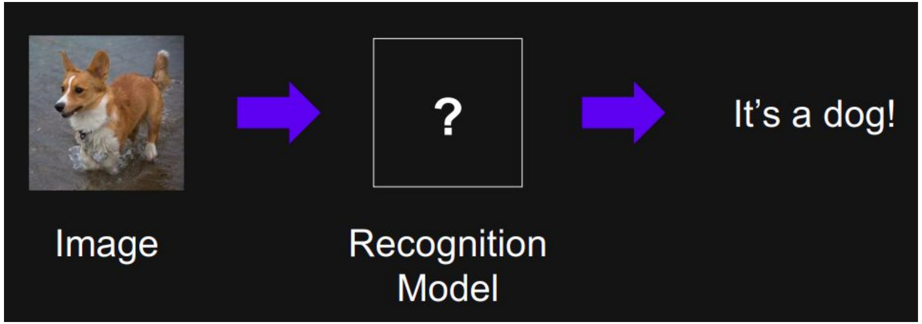
Mixup

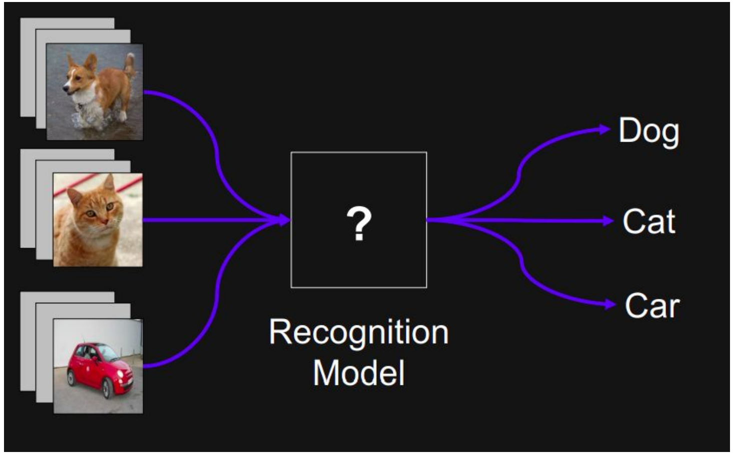
Manifold Mixup

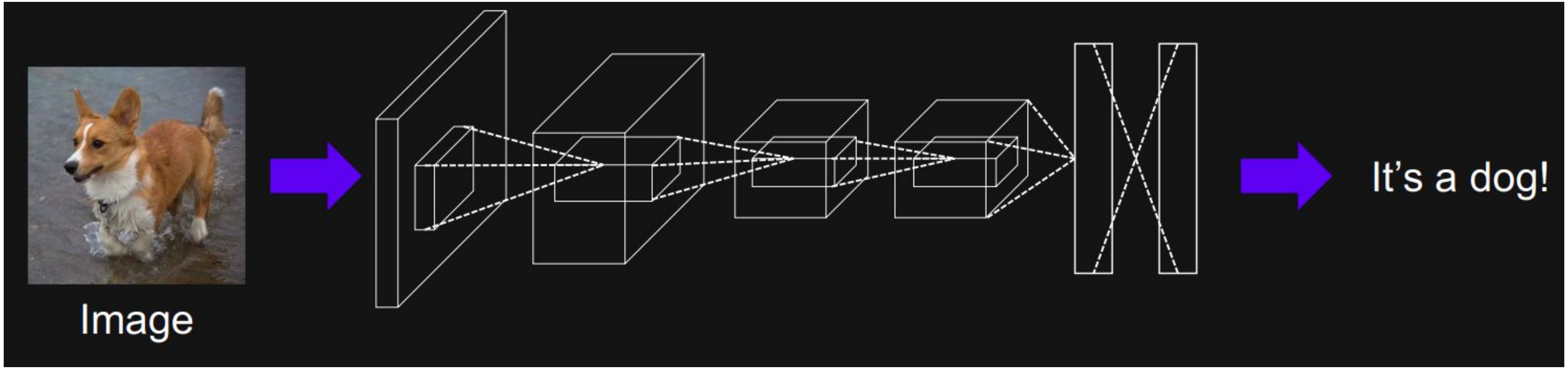
Cutmix

Data Augmentation

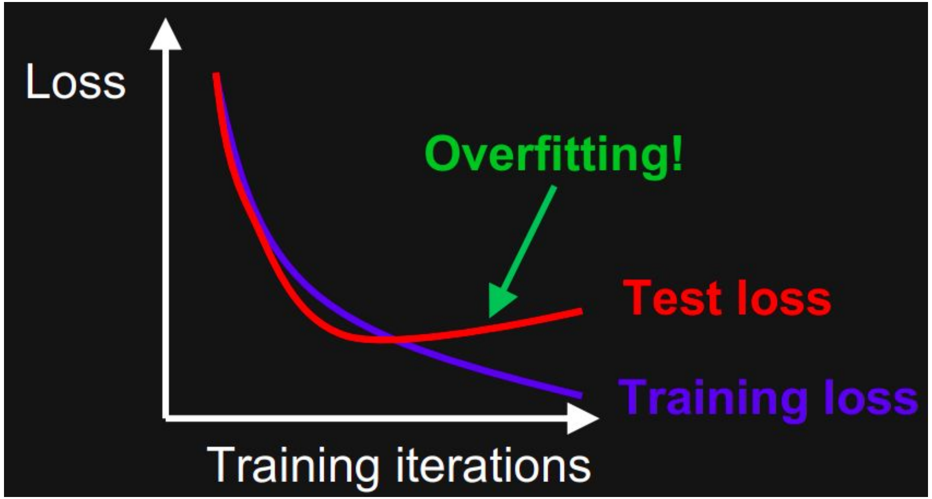
Data Augmentation

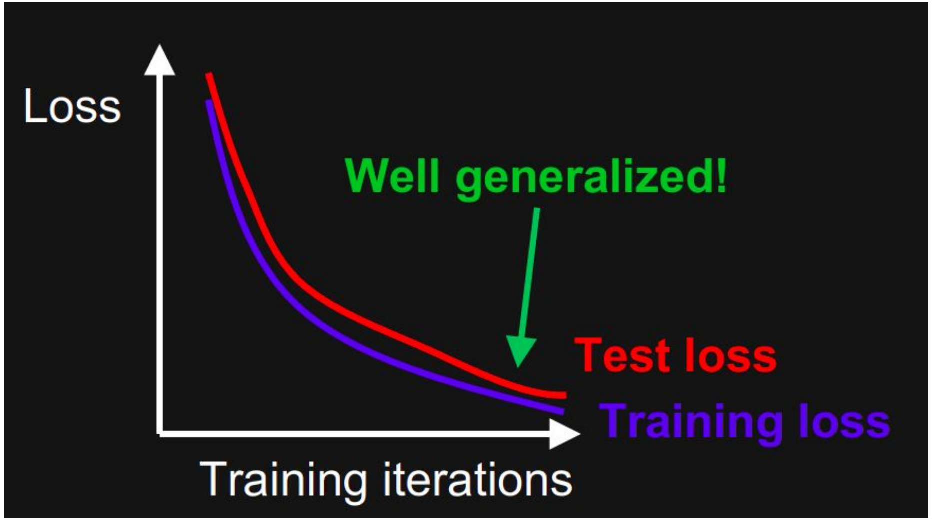






Data Augmentation



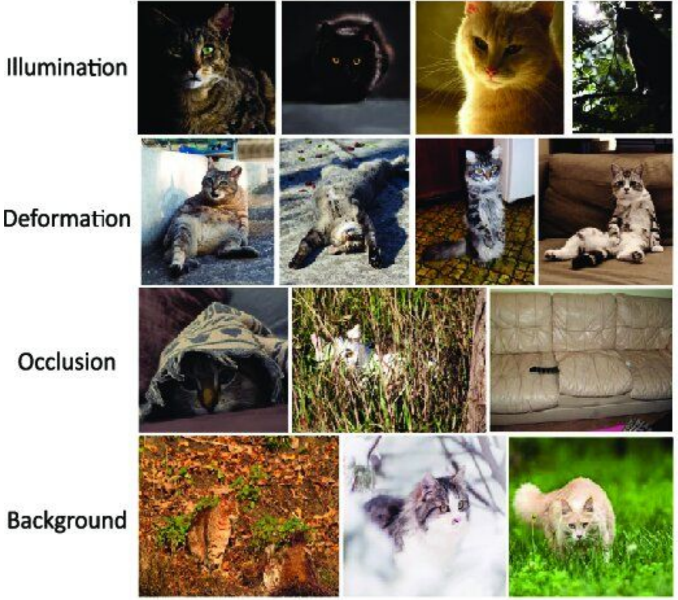


Data Augmentation

- **Data augmentation (DA)** is essential for ML
  - Increases the coverage of training data
  - Improves the generalizability of estimators
- An example of DA in the image domain:

goal is to maximize the performance using the same model & same dataset

Data Augmentation



Mixup

Mixup

Image $0.7 \times$  $+0.3 \times$  $=$ 

Label

[1.0, 0.0]
cat dog

[0.0, 1.0]
cat dog

[0.7, 0.3]
cat dog

Figure: Mixup for Image Classification

Mixup formulation

With coefficient $\lambda \sim \mathrm{Beta}(\alpha, \alpha)$, for $\lambda \in [0, 1], \alpha \in (0, \infty)$.
Mixup generates a virtual in-between sample,

$$\tilde{x} = \lambda x_i + (1 - \lambda) x_j,$$

$$\tilde{y} = \lambda y_i + (1 - \lambda) y_j,$$

The mixup hyper-parameter α controls the strength of interpolation between feature-target pair

Smoother feature space

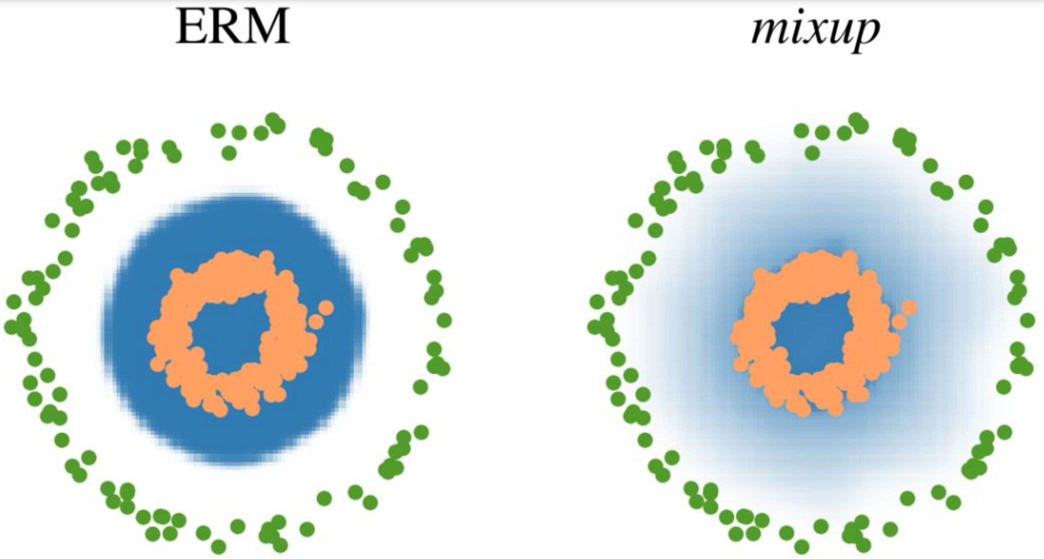
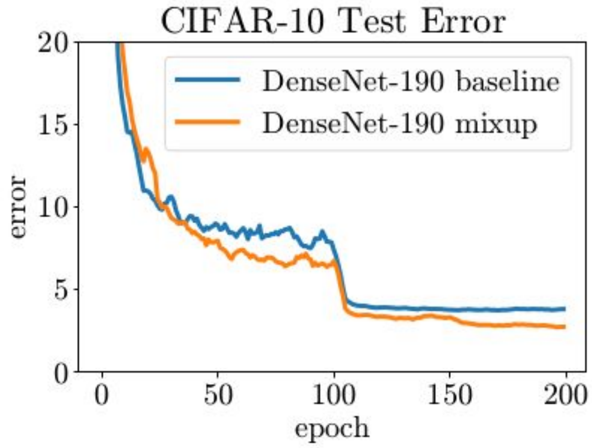


Figure: Illustrative sample referred from [Zhang et al., 2018]. The green and orange dots represent different classes. Blue shading indicates the probability $p(y = 1|x)$. Mixup yields a smoother decision boundary in feature space than ERM.

Mixup

Dataset	Model	ERM	*mixup*
CIFAR-10	PreAct ResNet-18	5.6	**4.2**
	WideResNet-28-10	3.8	**2.7**
	DenseNet-BC-190	3.7	**2.7**
CIFAR-100	PreAct ResNet-18	25.6	**21.1**
	WideResNet-28-10	19.4	**17.5**
	DenseNet-BC-190	19.0	**16.8**

(a) Test errors for the CIFAR experiments.



(b) Test error evolution for the best ERM and *mixup* models.

Figure 3: Test errors for ERM and *mixup* on the CIFAR experiments.

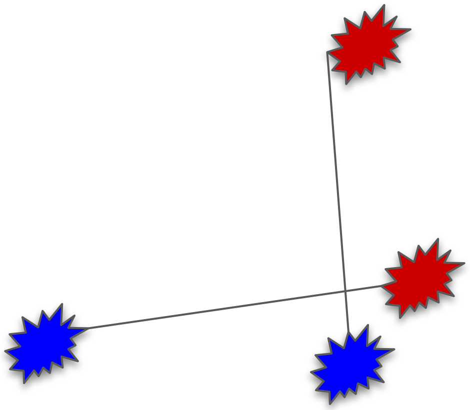
Manifold Mixup

Manifold Mixup

- Train on hidden states which are randomly interpolated between examples.

- Then train these interpolated hidden states to lead to lower confidence outputs.

- This also forces the model to learn representations which permit consistent interpolations.

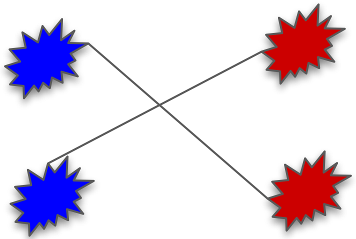


90% blue,
10% red

vs.

10% blue,
90% red

High error!



50% red,
50% blue

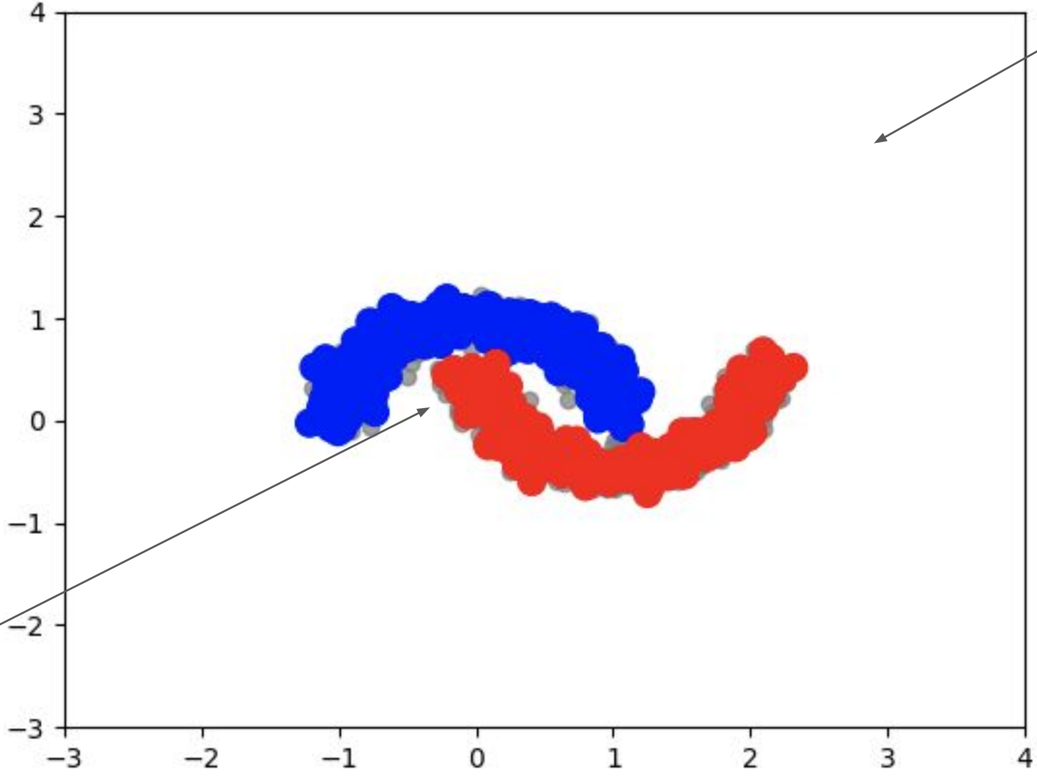
vs.

50% blue,
50% red

Low error!

Manifold Mixup

Should be low confidence, since there's no data here.



Should be low confidence because it's pretty close to both classes.

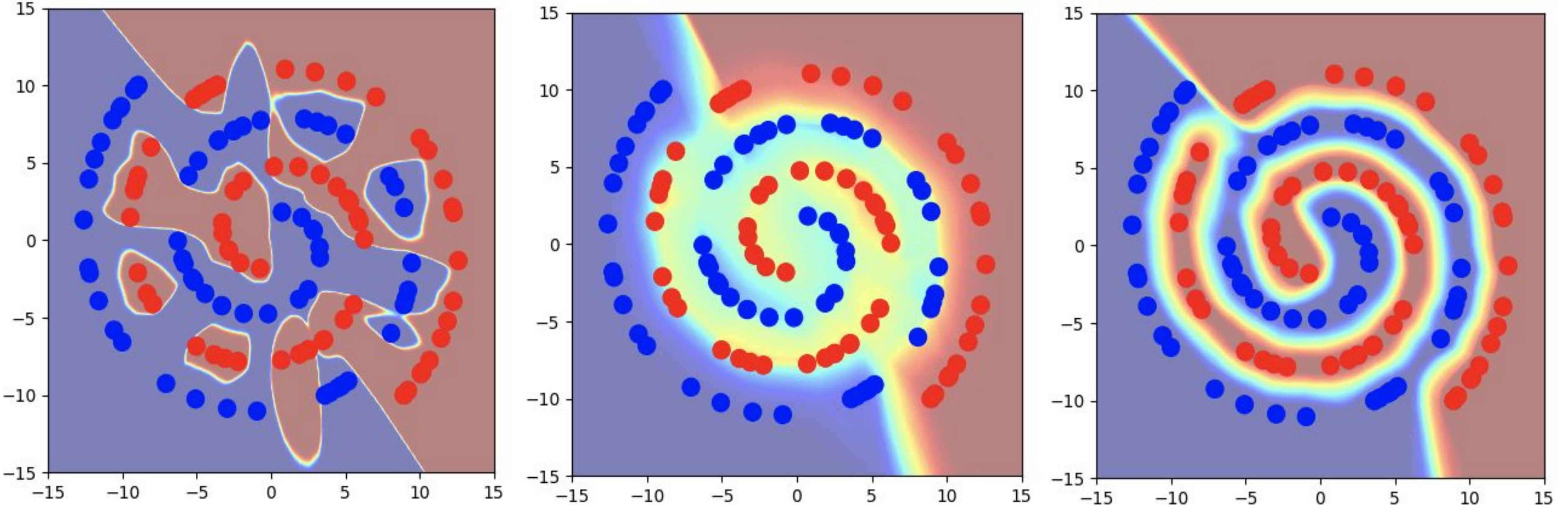
Manifold Mixup

- On each update, pick a random layer uniformly (including the input).

-Sample lambda ~ Beta(alpha, alpha)

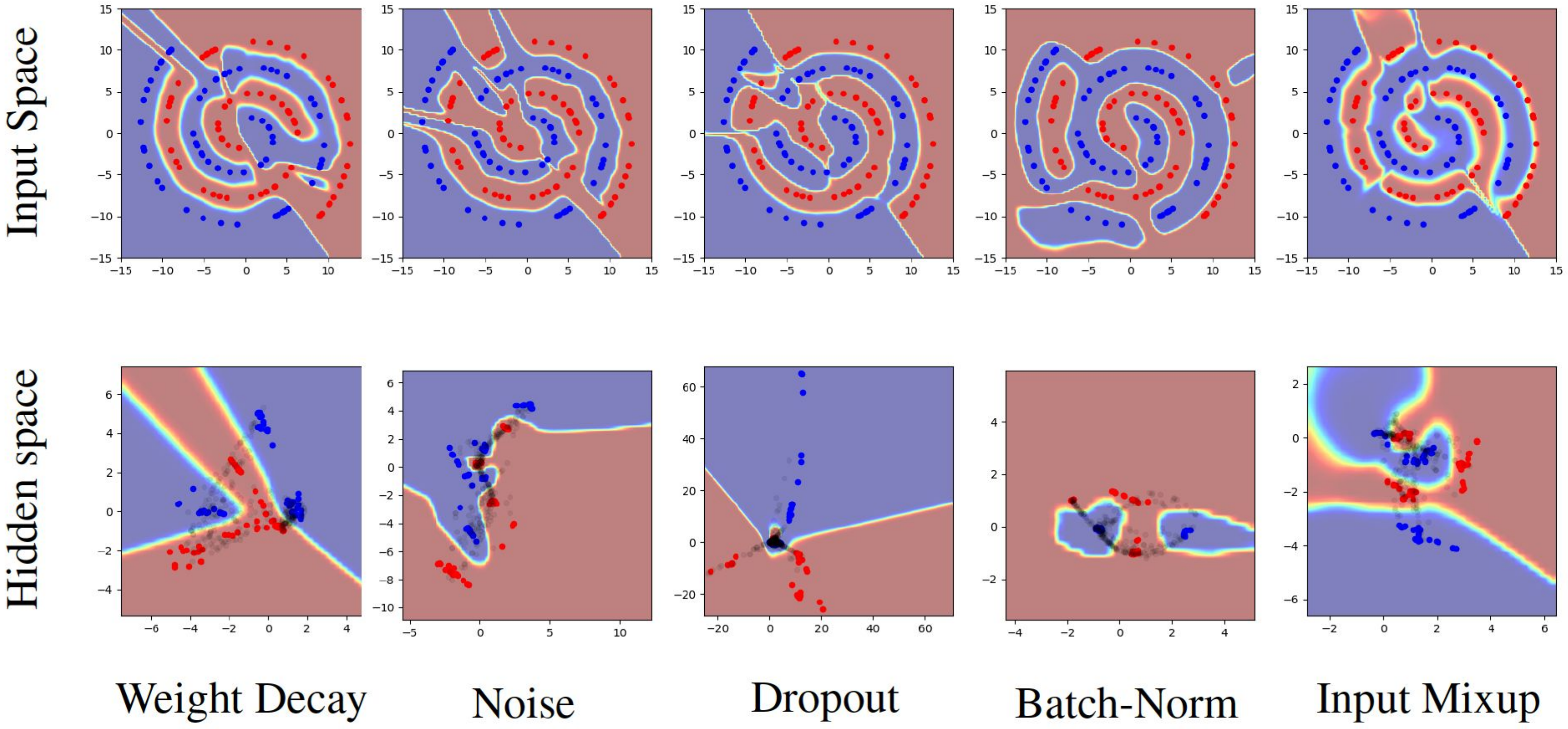
-Mix between two random examples from the minibatch at that layer with rate lambda.

-Mix the labels for those two examples accordingly (soft label).



None | Input Mixup | Manifold Mixup

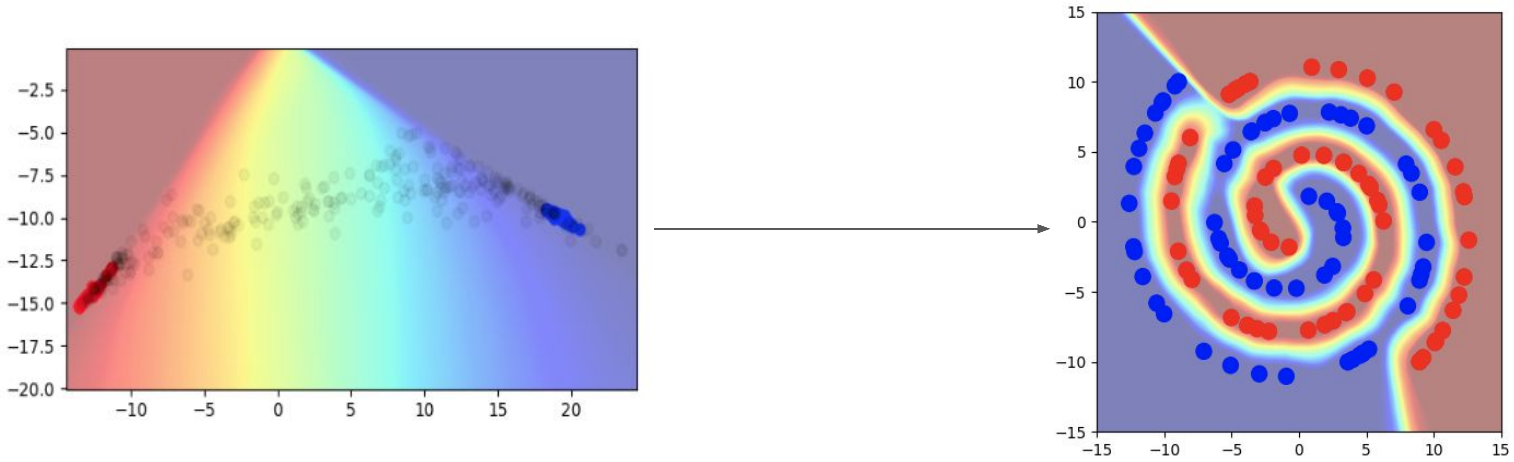
Manifold Mixup



Manifold Mixup

-Encourage most of the hidden space to correspond to low confidence classifications.

-Encourage real data's representations to be concentrated into local regions.



Manifold Mixup

Table 1: Classification errors on (a) CIFAR-10 and (b) CIFAR-100. We include results from (Zhang et al., 2018)† and (Guo et al., 2016)‡. Standard deviations over five repetitions.

PreActResNet18	Test Error (%)	Test NLL
No Mixup	4.83 ± 0.066	0.190 ± 0.003
AdaMix‡	3.52	NA
Input Mixup†	4.20	NA
Input Mixup ($\alpha = 1$)	3.82 ± 0.048	0.186 ± 0.004
Manifold Mixup ($\alpha = 2$)	$\underline{2.95 \pm 0.046}$	$\underline{0.137 \pm 0.003}$
PreActResNet34		
No Mixup	4.64 ± 0.072	0.200 ± 0.002
Input Mixup ($\alpha = 1$)	2.88 ± 0.043	0.176 ± 0.002
Manifold Mixup ($\alpha = 2$)	$\underline{2.54 \pm 0.047}$	$\underline{0.118 \pm 0.002}$
Wide-Resnet-28-10		
No Mixup	3.99 ± 0.118	0.162 ± 0.004
Input Mixup ($\alpha = 1$)	2.92 ± 0.088	0.173 ± 0.001
Manifold Mixup ($\alpha = 2$)	$\underline{2.55 \pm 0.024}$	$\underline{0.111 \pm 0.001}$

(a) CIFAR-10

PreActResNet18	Test Error (%)	Test NLL
No Mixup	24.01 ± 0.376	1.189 ± 0.002
AdaMix‡	20.97	n/a
Input Mixup†	21.10	n/a
Input Mixup ($\alpha = 1$)	22.11 ± 0.424	1.055 ± 0.006
Manifold Mixup ($\alpha = 2$)	$\underline{20.34 \pm 0.525}$	$\underline{0.912 \pm 0.002}$
PreActResNet34		
No Mixup	23.55 ± 0.399	1.189 ± 0.002
Input Mixup ($\alpha = 1$)	20.53 ± 0.330	1.039 ± 0.045
Manifold Mixup ($\alpha = 2$)	$\underline{18.35 \pm 0.360}$	$\underline{0.877 \pm 0.053}$
Wide-Resnet-28-10		
No Mixup	21.72 ± 0.117	1.023 ± 0.004
Input Mixup ($\alpha = 1$)	18.89 ± 0.111	0.927 ± 0.031
Manifold Mixup ($\alpha = 2$)	$\underline{18.04 \pm 0.171}$	$\underline{0.809 \pm 0.005}$

(b) CIFAR-100

Manifold Mixup

Table 2: Classification errors and neg-log-likelihoods on SVHN. We run each experiment five times.

PreActResNet18	Test Error (%)	Test NLL
No Mixup	2.89 ± 0.224	0.136 ± 0.001
Input Mixup ($\alpha = 1$)	2.76 ± 0.014	0.212 ± 0.011
Manifold Mixup ($\alpha = 2$)	$\underline{2.27 \pm 0.011}$	$\underline{0.122 \pm 0.006}$
PreActResNet34		
No Mixup	2.97 ± 0.004	0.165 ± 0.003
Input Mixup ($\alpha = 1$)	2.67 ± 0.020	0.199 ± 0.009
Manifold Mixup ($\alpha = 2$)	$\underline{2.18 \pm 0.004}$	$\underline{0.137 \pm 0.008}$
Wide-Resnet-28-10		
No Mixup	2.80 ± 0.044	0.143 ± 0.002
Input Mixup ($\alpha = 1$)	2.68 ± 0.103	0.184 ± 0.022
Manifold Mixup ($\alpha = 2$)	$\underline{2.06 \pm 0.068}$	$\underline{0.126 \pm 0.008}$

Table 3: Accuracy on TinyImagenet.

PreActResNet18	top-1	top-5
No Mixup	55.52	71.04
Input Mixup ($\alpha = 0.2$)	56.47	71.74
Input Mixup ($\alpha = 0.5$)	55.49	71.62
Input Mixup ($\alpha = 1.0$)	52.65	70.70
Input Mixup ($\alpha = 2.0$)	44.18	68.26
Manifold Mixup ($\alpha = 0.2$)	<u>58.70</u>	<u>73.59</u>
Manifold Mixup ($\alpha = 0.5$)	57.24	73.48
Manifold Mixup ($\alpha = 1.0$)	56.83	73.75
Manifold Mixup ($\alpha = 2.0$)	48.14	71.69

Cutmix

Cutmix

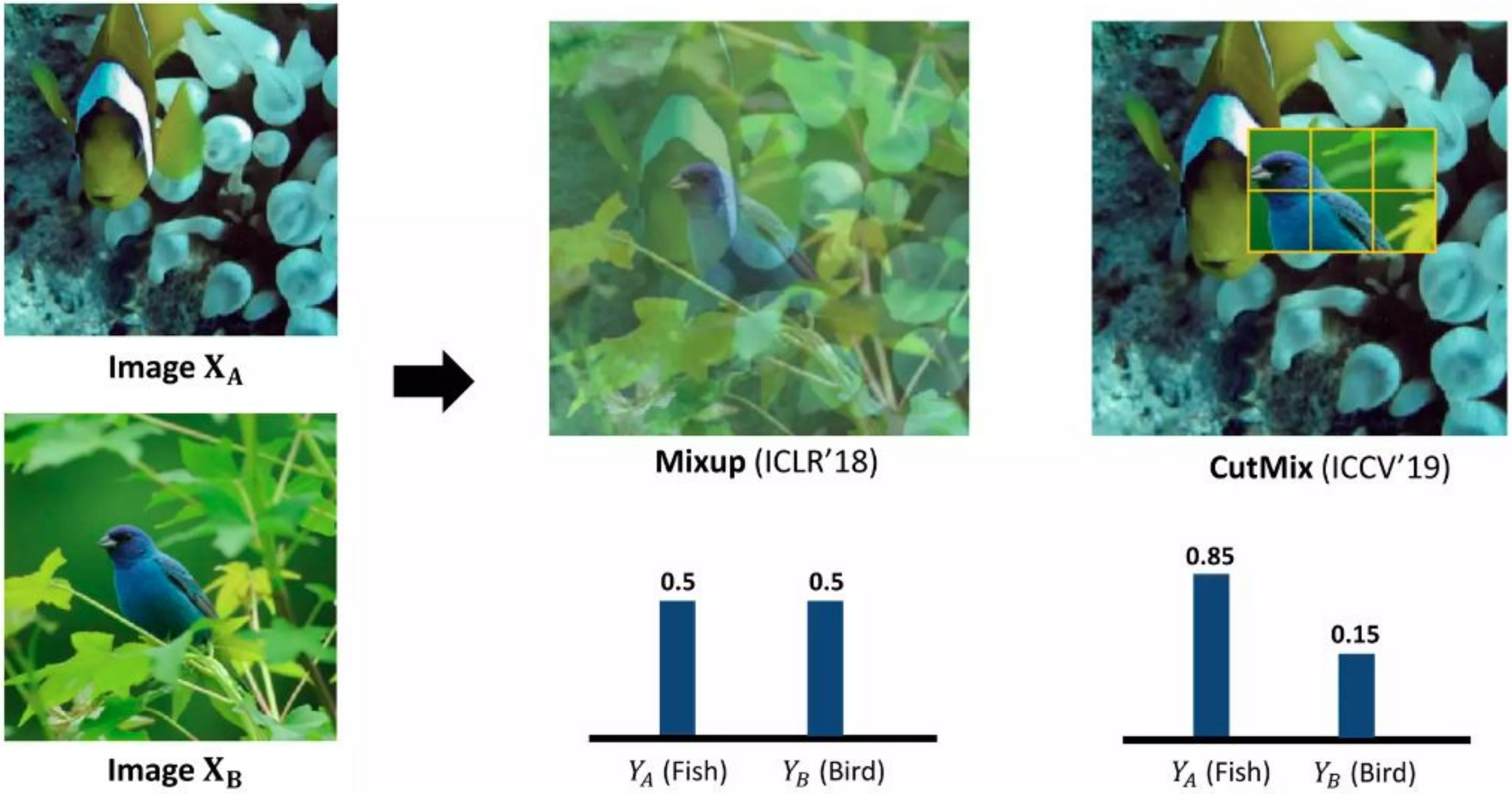


Image X_A
Image X_B
Mixup (ICLR'18)
CutMix (ICCV'19)
0.5
0.5
Y_A (Fish)
Y_B (Bird)
0.85
0.15
Y_A (Fish)
Y_B (Bird)

Cutmix

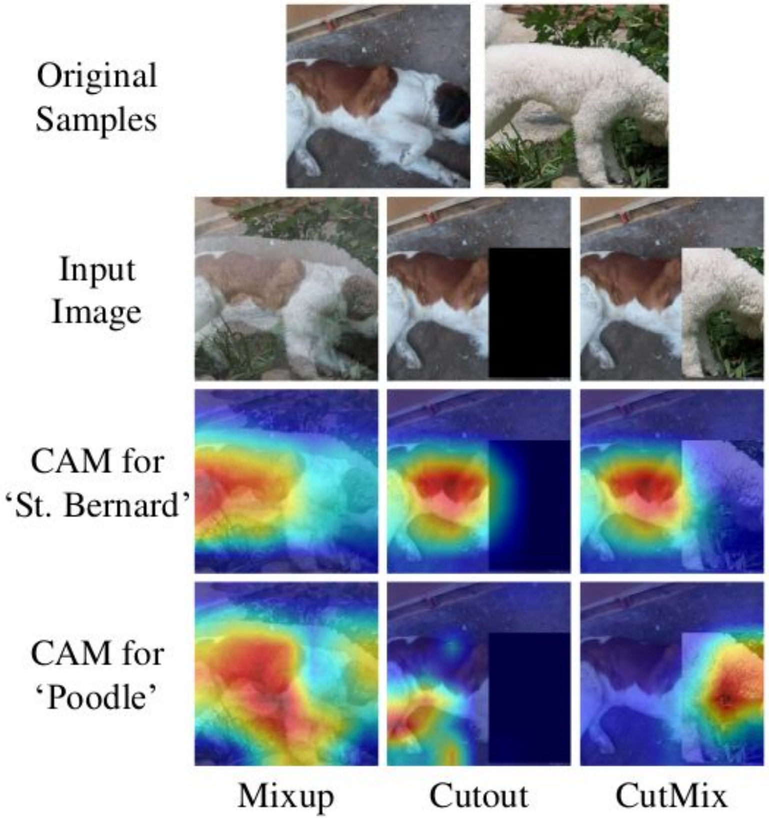
	ResNet-50	Mixup [48]	Cutout [3]	CutMix
Image				
Label	Dog 1.0	Dog 0.5 Cat 0.5	Dog 1.0	Dog 0.6 Cat 0.4
ImageNet Cls (%)	76.3 (+0.0)	77.4 (+1.1)	77.1 (+0.8)	**78.6 (+2.3)**
ImageNet Loc (%)	46.3 (+0.0)	45.8 (-0.5)	46.7 (+0.4)	**47.3 (+1.0)**
Pascal VOC Det (mAP)	75.6 (+0.0)	73.9 (-1.7)	75.1 (-0.5)	**76.7 (+1.1)**

Cutmix

Let $x \in \mathbb{R}^{W \times H \times C}$ and y denote a training image and its label, respectively. The goal of CutMix is to generate a new training sample $(\tilde{x}, \tilde{y})$ by combining two training samples (x_A, y_A) and (x_B, y_B). The generated training sample $(\tilde{x}, \tilde{y})$ is used to train the model with its original loss function. We define the combining operation as

$$\begin{aligned} \tilde{x} &= \mathbf{M} \odot x_A + (\mathbf{1} - \mathbf{M}) \odot x_B \\ \tilde{y} &= \lambda y_A + (1 - \lambda) y_B, \end{aligned} \tag{1}$$

$$r_x \sim \text{Unif}\ (0, W), \quad r_w = W\sqrt{1-\lambda},$$
$$r_y \sim \text{Unif}\ (0, H), \quad r_h = H\sqrt{1-\lambda}$$



Cutmix

PyramidNet-200 ($\tilde{\alpha}$=240) (# params: 26.8 M)	Top-1 Err (%)	Top-5 Err (%)
Baseline	16.45	3.69
+ StochDepth [17]	15.86	3.33
+ Label smoothing (ϵ=0.1) [38]	16.73	3.37
+ Cutout [3]	16.53	3.65
+ Cutout + Label smoothing (ϵ=0.1)	15.61	3.88
+ DropBlock [8]	15.73	3.26
+ DropBlock + Label smoothing (ϵ=0.1)	15.16	3.86
+ Mixup (α=0.5) [48]	15.78	4.04
+ Mixup (α=1.0) [48]	15.63	3.99
+ Manifold Mixup (α=1.0) [42]	16.14	4.07
+ Cutout + Mixup (α=1.0)	15.46	3.42
+ Cutout + Manifold Mixup (α=1.0)	15.09	3.35
+ ShakeDrop [46]	15.08	2.72
+ CutMix	14.47	2.97
+ CutMix + ShakeDrop [46]	**13.81**	**2.29**

Table 5: Comparison of state-of-the-art regularization methods on CIFAR-100.

Model	# Params	Top-1 Err (%)	Top-5 Err (%)
PyramidNet-110 ($\tilde{\alpha} = 64$) [11]	1.7 M	19.85	4.66
PyramidNet-110 + CutMix	1.7 M	**17.97**	**3.83**
ResNet-110 [12]	1.1 M	23.14	5.95
ResNet-110 + CutMix	1.1 M	**20.11**	**4.43**

Table 6: Impact of CutMix on lighter architectures on CIFAR-100.

PyramidNet-200 ($\tilde{\alpha}$=240)	Top-1 Error (%)
Baseline	3.85
+ Cutout	3.10
+ Mixup (α=1.0)	3.09
+ Manifold Mixup (α=1.0)	3.15
+ CutMix	**2.88**

Table 7: Impact of CutMix on CIFAR-10.

Feature/Technique	Mixup	Manifold Mixup	CutMix
Basic Concept	Combines two or more input images and their labels linearly.	Similar to Mixup, but mixes hidden representations at various layers of the network.	Cuts and pastes patches from one image onto another, mixing the labels accordingly.
Data Augmentation	Operates at the input level (pixel values).	Operates at both input and hidden layers within the network.	Operates at the input level with a focus on spatial regions.
Primary Goal	Encourages linear behavior between training examples.	Encourages learning more robust features across different tasks.	Aims at improving localization and understanding of spatial context.
Label Mixing	Labels are mixed in a linear fashion according to the mix ratio.	Labels are mixed based on the level at which mixing occurs.	Labels are mixed proportionally to the area of the patches involved.
Image Mixing	Linear interpolation of pixel values.	Interpolation of features at different network layers.	Physical combination of image patches.
Impact on Training	Helps in generalizing to unseen data by smoothing the decision boundary.	Promotes learning of more generalizable and robust intermediate features.	Enhances the ability of the model to localize and recognize objects within a varied context.
Use Cases	Generally used in image classification tasks.	Useful in tasks requiring deeper feature understanding and abstraction.	Particularly beneficial in object detection and classification tasks.

Technique	Advantages	Disadvantages
Traditional Data Augmentation	- Realistic modifications (rotation, flipping, scaling) - Simple to implement - Improves generalization and reduces overfitting	- Limited to predefined variations - Can be computationally inefficient - May not represent actual data distribution
Mixup	- Enhances regularization, favoring linear behavior between training examples - Improves generalization to unseen data - Robustness to label noise	- Generates potentially unrealistic synthetic samples - Complexity in interpretation of mixed images and labels
Manifold Mixup	- Encourages learning of abstract and robust features - Versatile (applicable at multiple network layers) - Potentially better regularization than Mixup	- More complex implementation - Additional computational overhead - Risk of over-smoothing decision boundaries
CutMix	- Enhances spatial context learning and object localization - Robust to occlusion - Balanced regularization (mix of dropout and Mixup)	- Can introduce artificial artifacts - Complex label handling based on patch area - Risk of feature misalignment in cut-and-paste