CS24

2019.2.12
修士論文公聴会

Data Augmentation using Random Image Cropping and Patching for Deep CNNs

深層畳み込みニューラルネットワークのための、
ランダムな画像くり抜きと貼り付けを用いた
data augmentationの提案

計算科学専攻 CS24

171x214x 高橋良

指導教員 上原邦昭教授

研究背景｜深層学習による画像識別

[画像識別タスク]

- データとして与えられた画像が何であるかを識別するタスク

入力画像

?

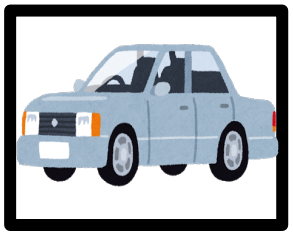
“car”

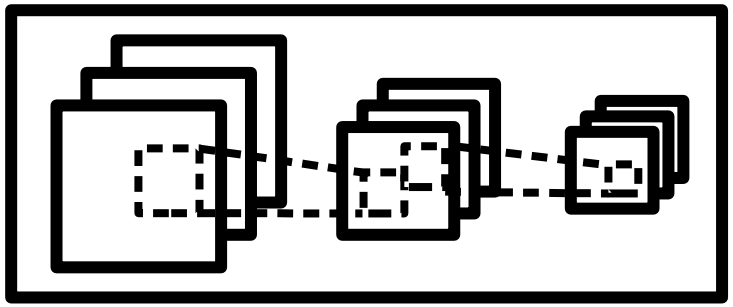
出力（クラス等）

例

（車、船、飛行機、人間）
のうちどれか

[深層学習による手法]





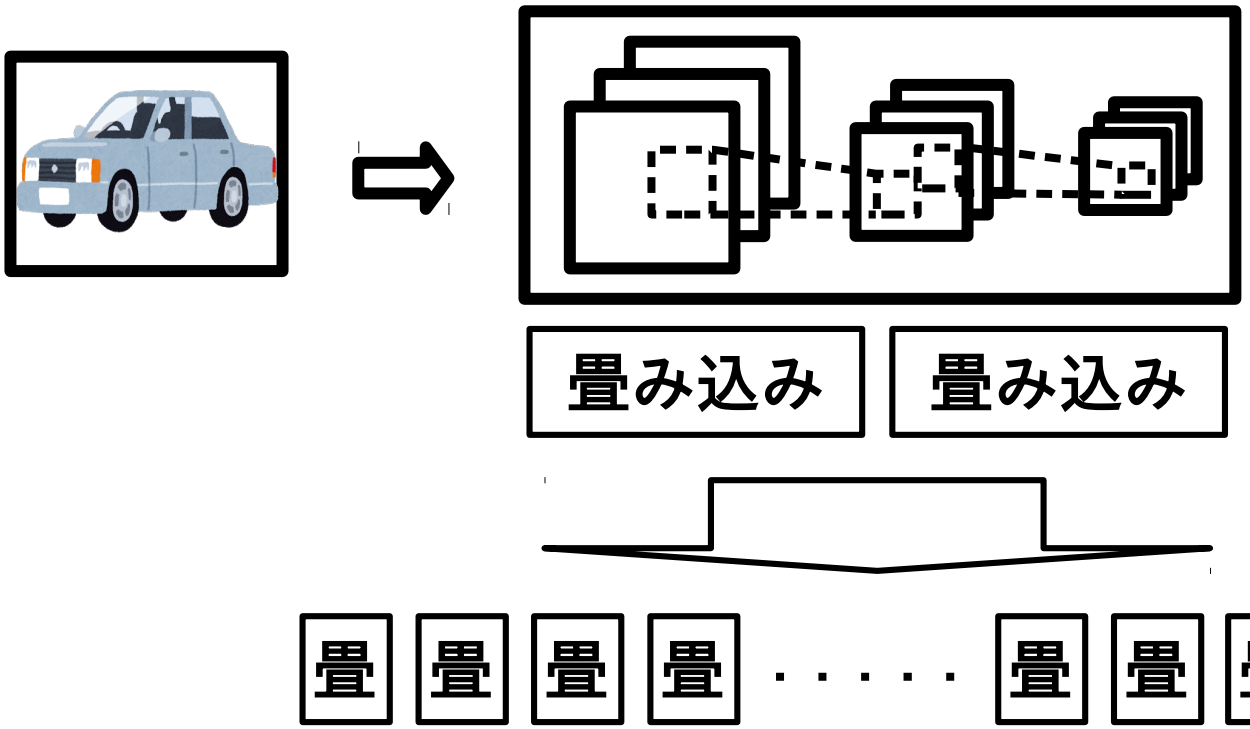
畳み込みニューラルネットワーク(CNN)
（フィルターを出力から**自動で学習**）

“car”

出力（クラス等）

研究背景 | 深層学習による画像識別

[CNNの多層化]



2～5 層程度

100～1000 層

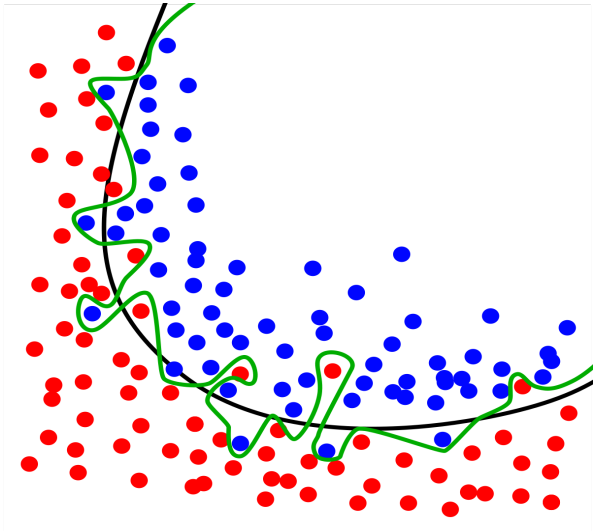
⇨ 計算機の進化、学習アルゴリズムの発展による多層化実現

⇨ **パラメータ数**が増え、さらなる**精度向上**が可能

研究背景｜深層学習による画像識別

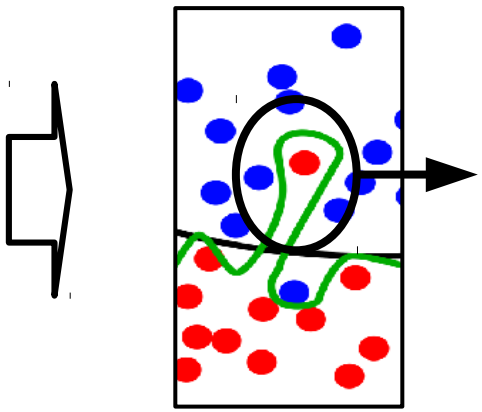
[多層の問題点：過学習]

- 学習に用いるデータに極端に適応してしまい、
  テストデータでの検証時ににかえって精度が悪化する
- **データ数、種類に対してパラメータ数が多いため生じる**



求めている境界線

過学習した境界線

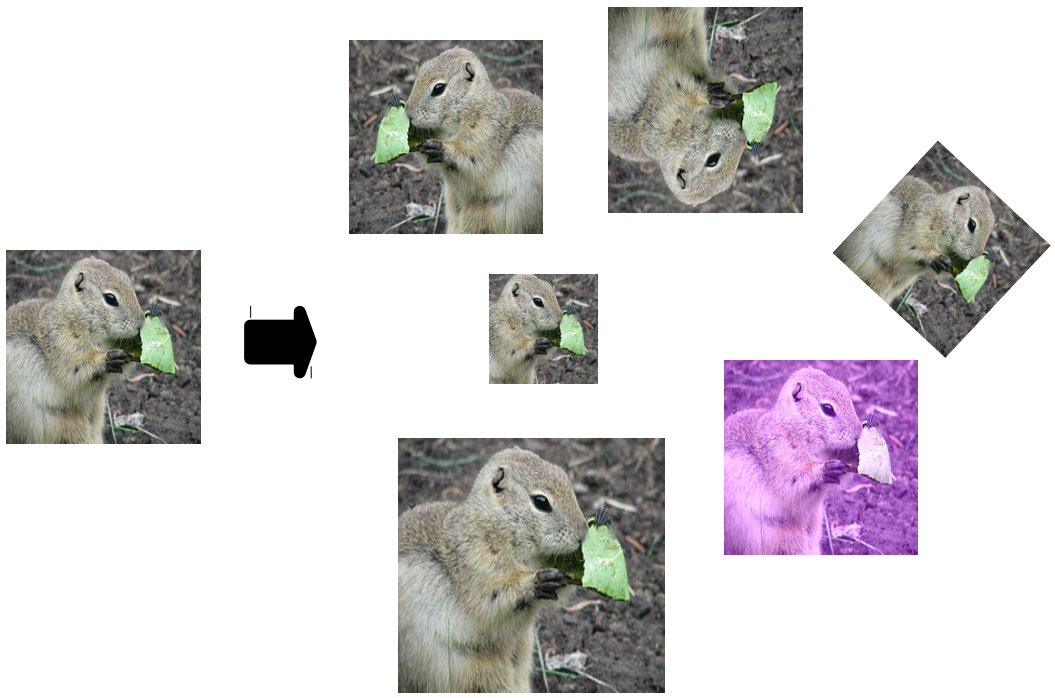


研究背景｜深層学習による画像識別

[Data Augmentation]

- 今手元にあるデータに変換処理を行い、擬似的に新しいデータを作り、**データ数、種類を拡張する ⇨ 過学習の抑制**
- 画像に対しては、基本的な画像処理を用いて行われる

・画像反転
・拡大、縮小
・回転
・くり抜き
・色彩変換



研究背景｜深層学習による画像識別

[Data Augmentationの効果]

	パラメータ数	テスト誤差 (%) data augmentation なし	テスト誤差 (%) data augmentation あり
モデル A	7.0M	5.77	4.10
モデル B	27.2M	5.83	3.74

[Huang, 2017, cvpr]

⇨ パラメータ数を増やし、順当に精度を向上させる上で
data augmentationは重要な役割を持つ

本研究：

**新規のdata augmentationを開発し
画像識別精度の更なる向上を目指す**

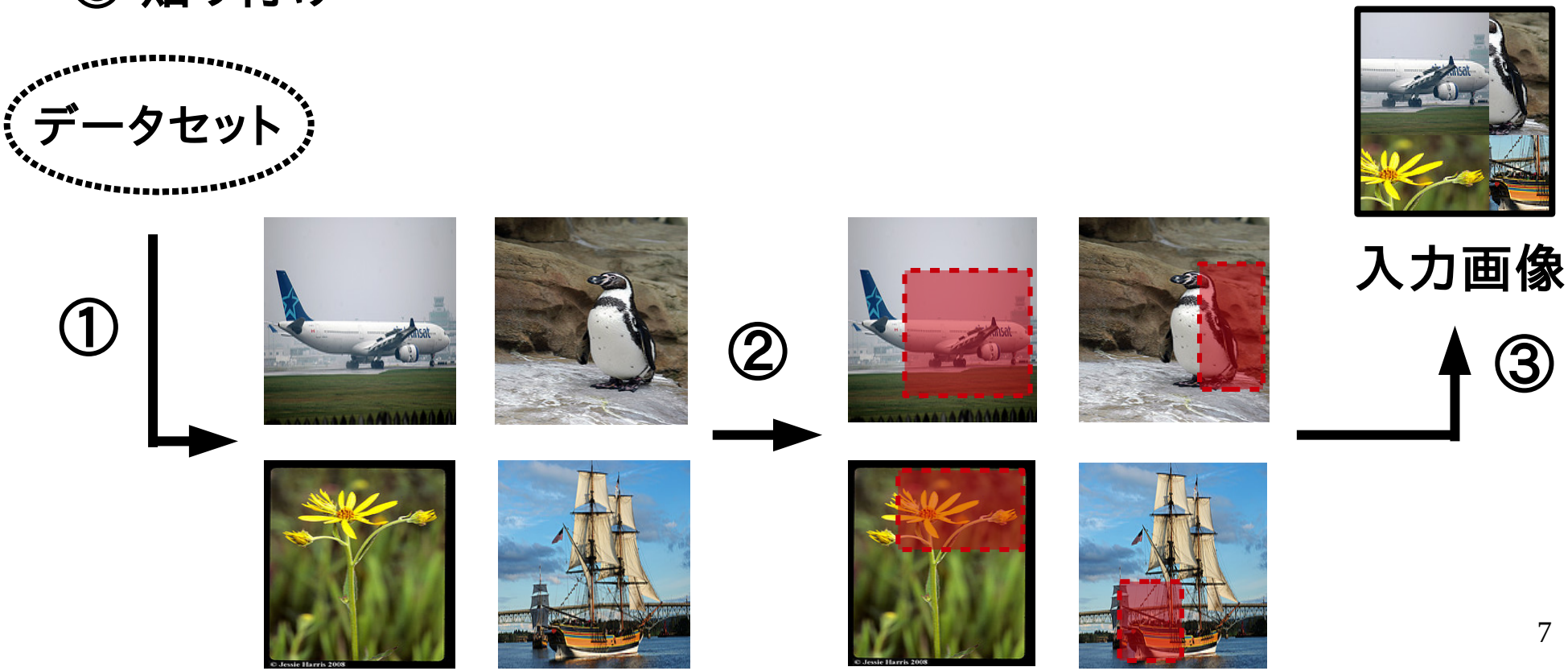
提案手法

[RICAP : random image cropping and patching]

① ランダムに4枚の画像を選択

② それぞれの画像の一部分をくり抜き

③ 貼り付け

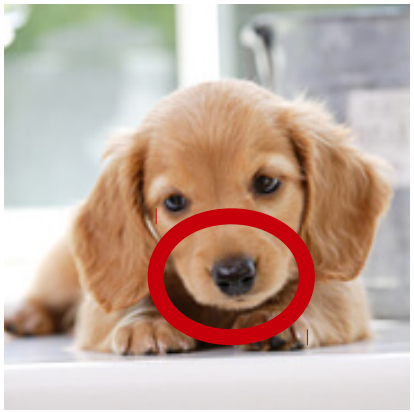


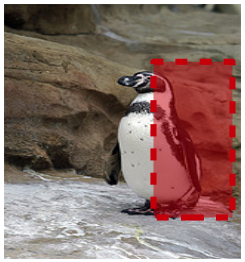
提案手法：RICAP

ポイント

- 画像の一部くり抜き
  - 学習毎に異なる箇所をくり抜くことで、
    モデルが注目し、学習する特徴を毎回変えることが出来る
    => **特定の特徴への過学習を防ぐ**

[例] 犬と猫を識別する時に、鼻の特徴だけ学習すれば
十分なとき、その他の特徴は学習されない

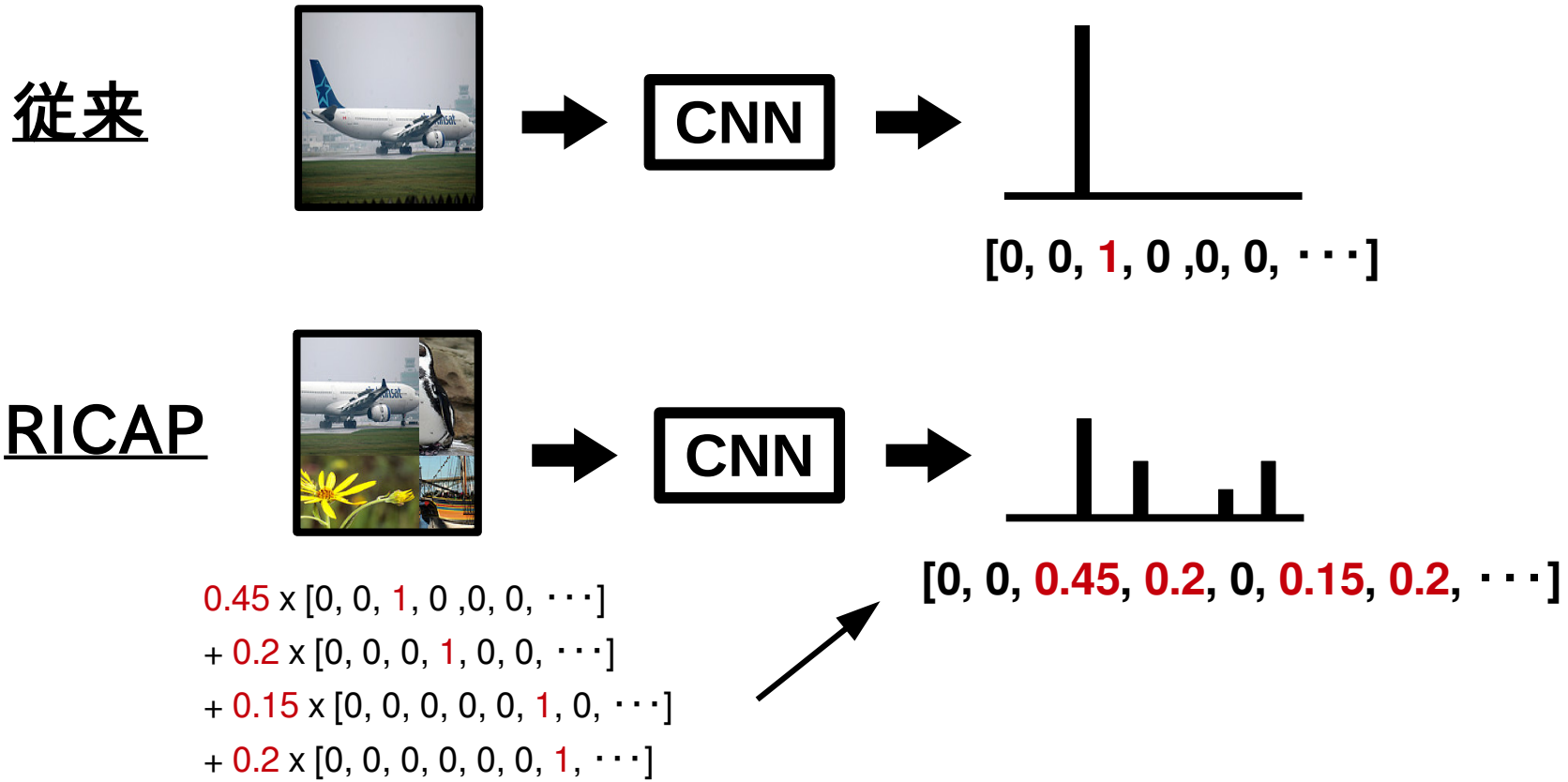




提案手法：RICAP

ポイント

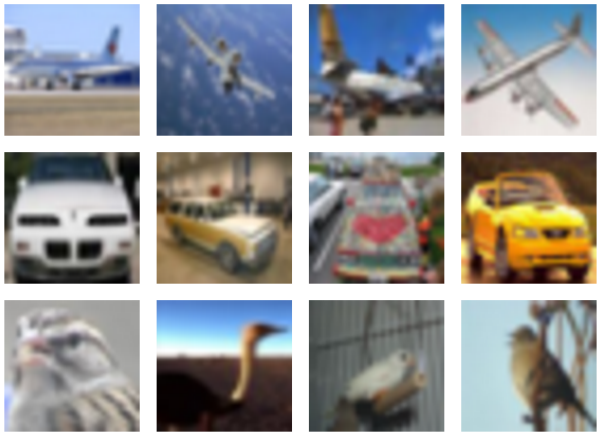
- 画像の貼り付け
  - クラスラベルのミックスによるタスクの複雑化
    => **学習を容易に収束させない正則化**



実験と結果 | データセット

[CIFAR-10 / CIFAR-100]

- 10 / 100 クラスのカラー画像 (飛行機、自動車、鳥、etc)
- 画像サイズ： 縦 x 横 x チャネル => 32 x 32 x 3
- 訓練データ　: 50,000枚
- テストデータ : 10,000枚



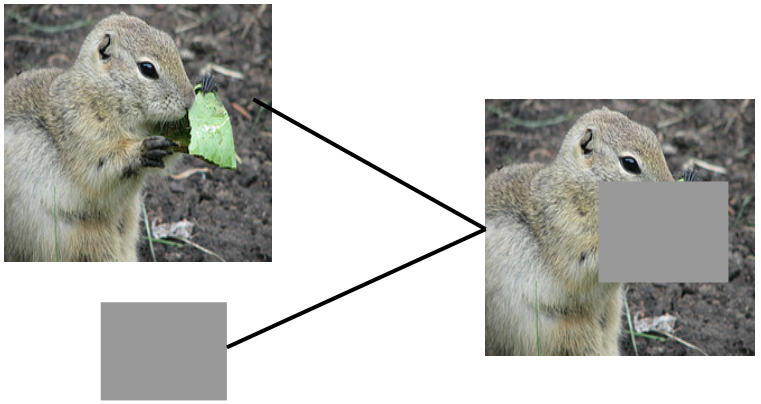
[ImageNet]

- 1000 クラスのカラー画像
- 画像サイズ(入力) : 縦 x 横 x チャネル => 256 x 256 x 3
- 訓練データ　: 128万枚
- テストデータ : 50,000枚

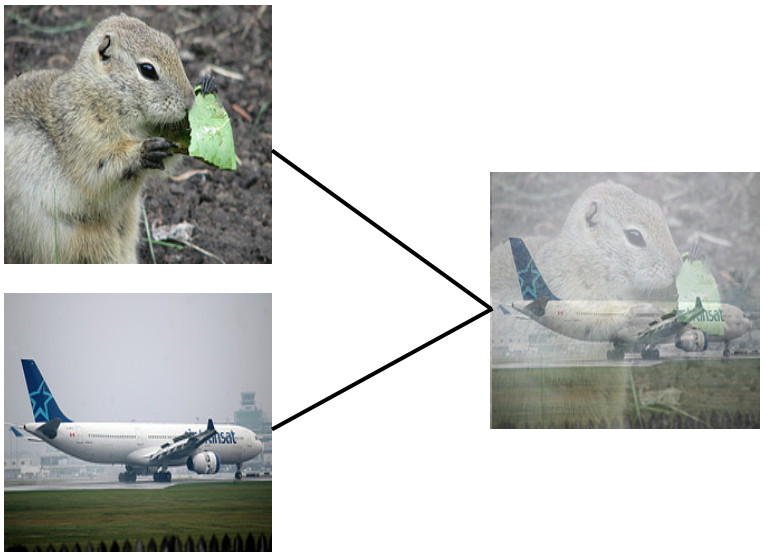
IMAGENET

実験と結果 | 先行（比較）研究

- **Cutout** [Devries+, arXiv, 2017]
  - 画像の矩形領域のマスク
  - 特定の特徴への過学習抑制



- **Mixup** [Zhang+, ICLR, 2018]
  - 2枚の画像のアルファブレンド
  - 画像をまたいで特徴を多様に



実験と結果 | 識別精度

[先行研究との比較]

- データセット : **CIFAR-10 / 100**
- 深層CNNモデル : **Wide ResNet** [Zagoruyko+, BMVC, 2016]

Method	CIFAR-10 Error(%)	CIFAR-100 Error(%)
Baseline	6.97	26.06
standard data augmentation	3.89	18.85
+ cutout	3.08 ±0.16	18.41 ±0.27
+ mixup	3.02 ±0.04	17.62 ±0.25
+ RICAP(ours)	**2.85** ±0.06	**17.22** ±0.20

実験と結果 | 識別精度

[モデルに対する汎用性]

- データセット : **CIFAR-10**
- 深層CNNモデル : Wide ResNet以外の**3つの深層CNNモデル**
  - DenseNet [Huang+, CVPR, 2017]
  - Pyramidal ResNet [Han+, CVPR, 2017]
  - ShakeShake ResNet [Gastaldi, ICLR, 2017]

	Error(%)		
Method	DenseNet	Pyramidal ResNet	ShakeShake
standard data augmentation	3.46	3.31 ±0.08	2.86
+ cutout	2.73 ±0.06	2.84 ±0.05	2.56 ±0.07
+ mixup	2.73 ±0.08	2.57 ±0.09	2.32 ±0.11
+ RICAP(ours)	**2.69** ±0.12	**2.51** ±0.02	**2.19** ±0.08

実験と結果 | 識別精度

[大規模データセットによる評価]

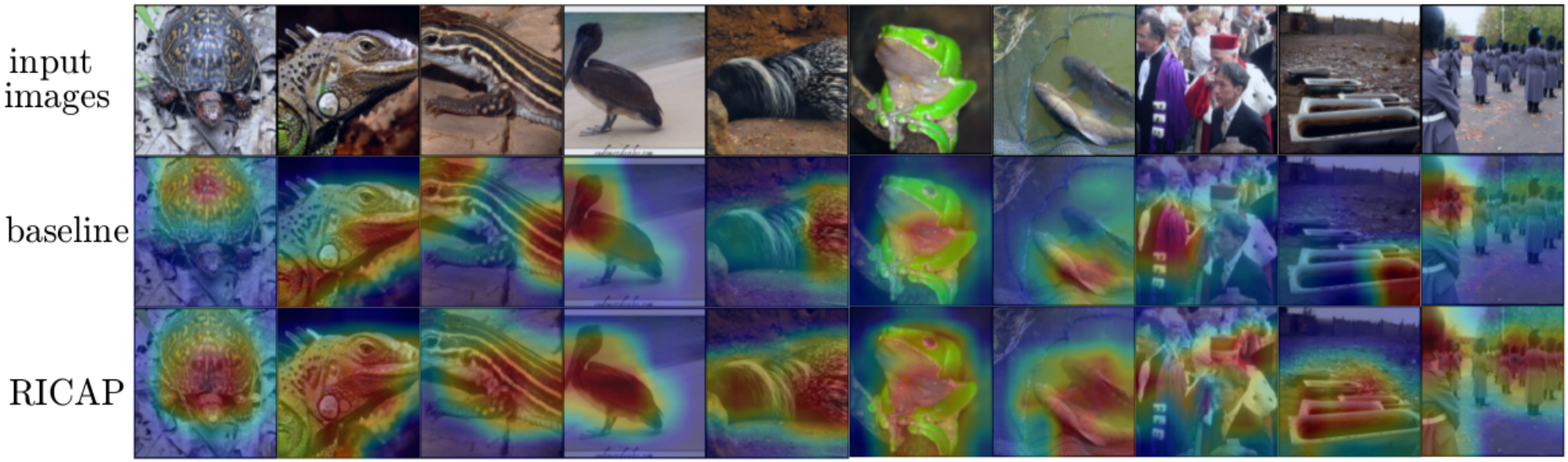
- データセット : **ImageNet**
- 深層CNNモデル : **Wide ResNet**

Method	top-1 Error(%)	top-5 Error(%)
+ standard data augmentation	21.90	6.03
+ cutout	22.45	6.22
+ mixup	21.83	5.81
+ RICAP(ours)	**21.08**	**5.66**

実験と結果｜定性的評価

[CNNの注目箇所の可視化]

- 提案手法：画像のくり抜きと貼り付けの処理によって、
  各画像のより多様な特徴の学習を期待
- モデルが識別時に注目している箇所を可視化して検証



class activation mapping [Zhou, cvpr, 2016]

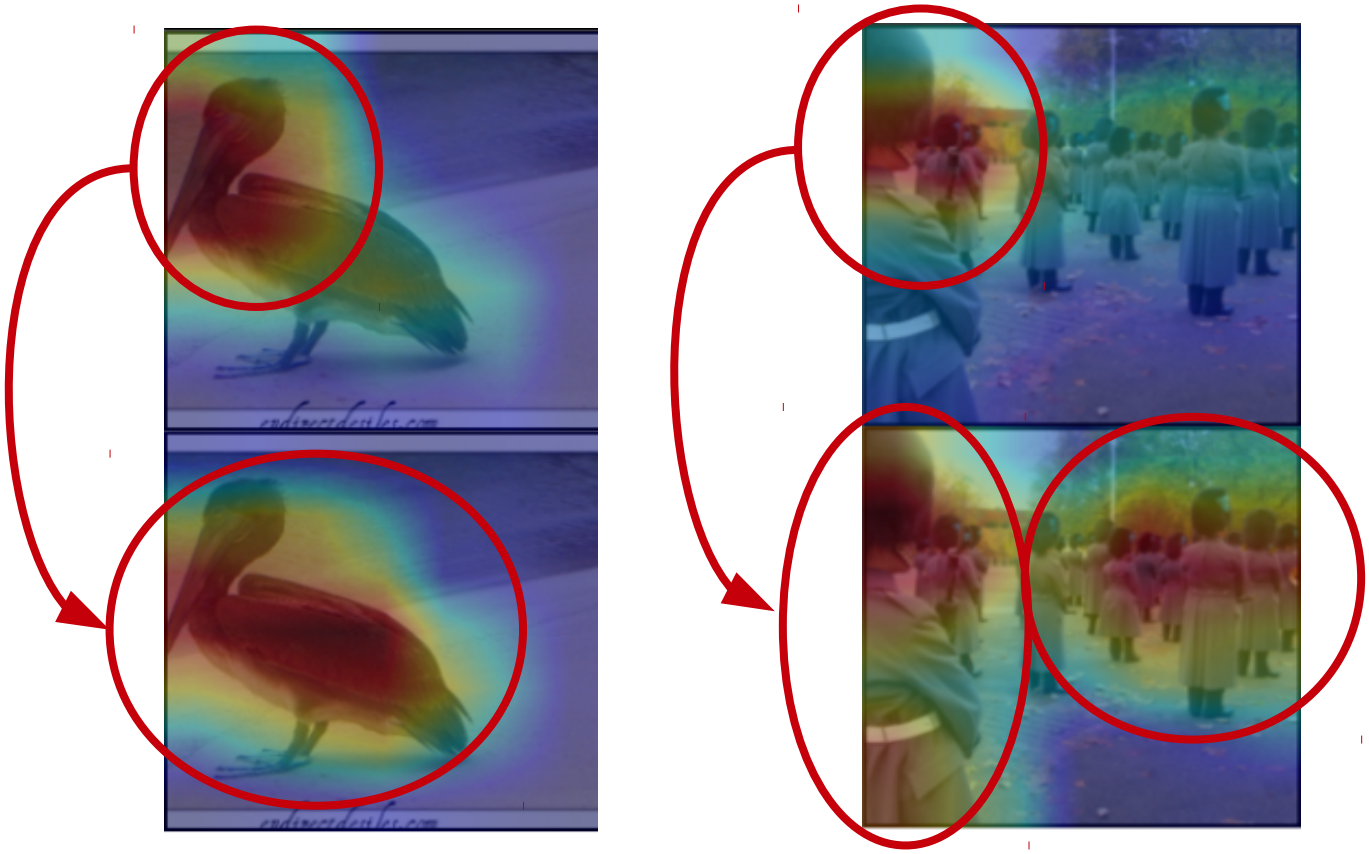
実験と結果｜定性的評価

[CNNの注目箇所の可視化]

例）

提案手法**なし**

提案手法**あり**



⇨ 提案手法を用いることで、特定箇所の特徴に引っ張られず、
より多様な特徴を捉えることが可能になっている

実験と結果 | まとめ

- **実験結果から**
  - ▶ 識別精度の向上

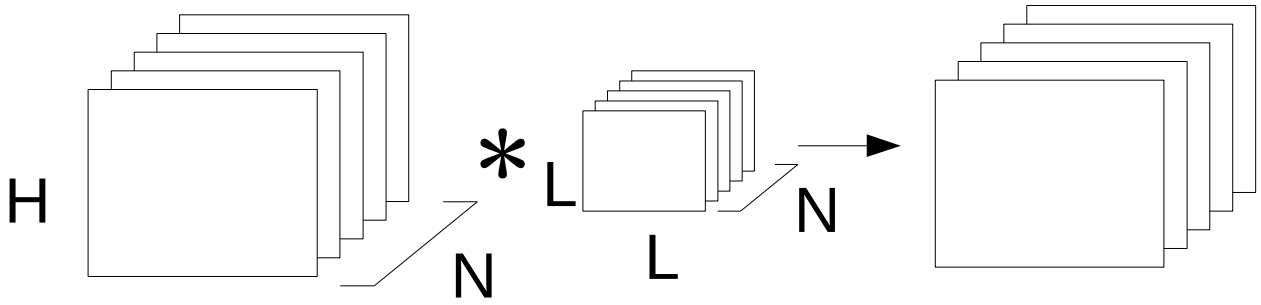
- **今後の課題**
  - ▶ より多くのCNNモデルで実験
  - ▶ 画像識別以外のタスクへの応用

} 汎用性追求

付録

研究背景 | 画像畳み込み

画像に対する畳み込み処理



x：入力画像

w：フィルタ

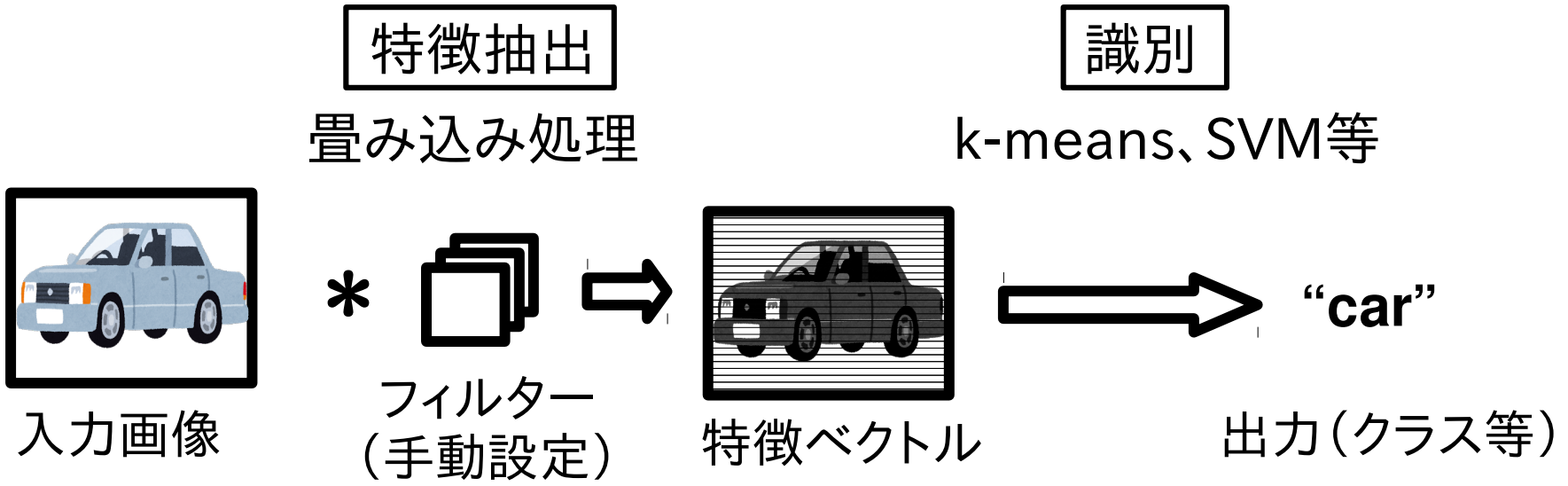
y：出力

$$y_{ijk} = \sum_{m=0}^{L-1} \sum_{n=0}^{L-1} w_{mnk} x_{(i+m)(j+n)k} + b_k$$

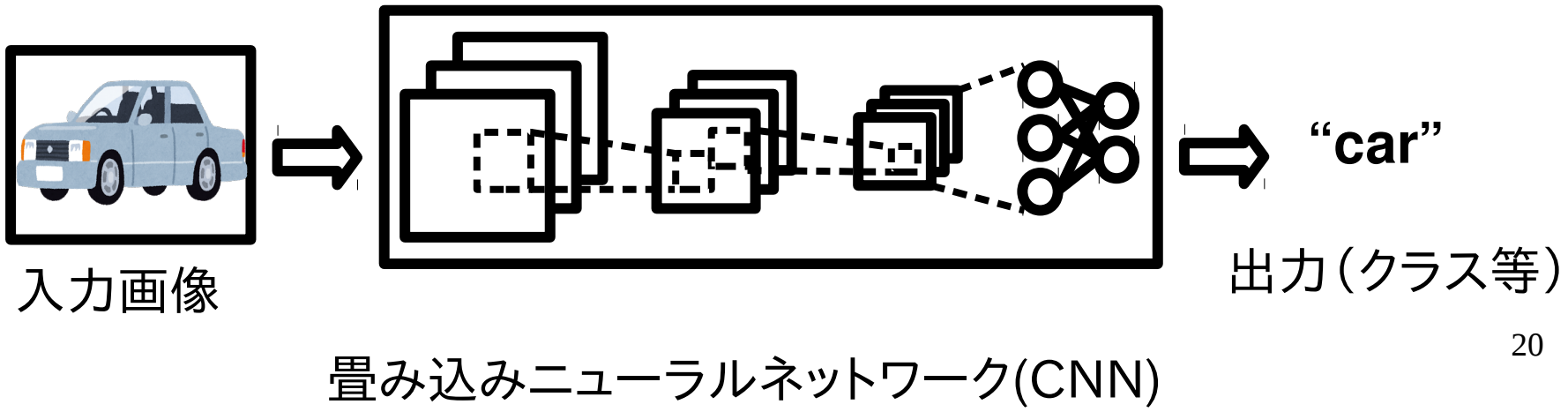
ijk：　画像の縦、横、チャネル

研究背景｜深層学習による画像識別

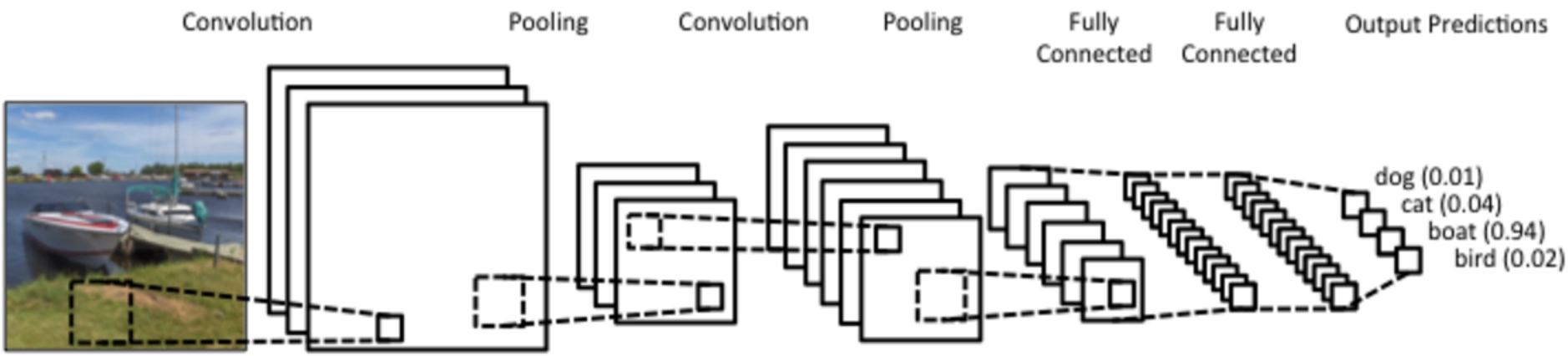
[従来の画像識別]



[深層学習]



研究背景 | CNN



畳み込み　畳み込み　畳み込み　畳み込み　識別

$$y_{ijk} = \sum_{m=0}^{L-1} \sum_{n=0}^{L-1} w_{mnk} x_{(i+m)(j+n)k} + b_k$$

⇨ **w**：フィルタの値がネットワークのパラメータ(重み)となって学習される

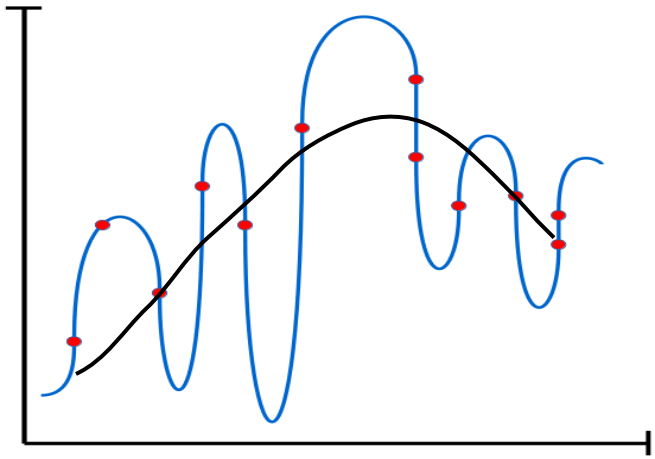
研究背景｜深層学習による画像識別

[多層の問題点]

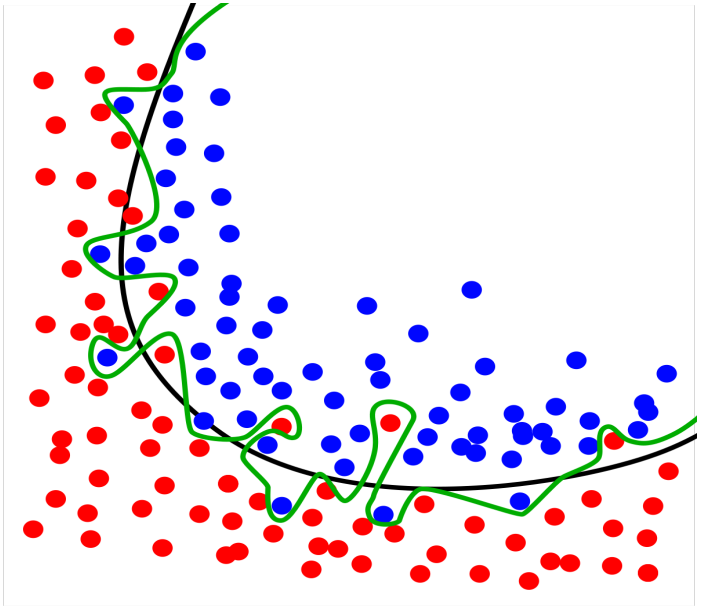
- パラメータが極端に多いために、**過学習**を起こしやすい

[過学習]

- 学習データに対して極端に適応した学習をしてしまい、汎化性を失うこと



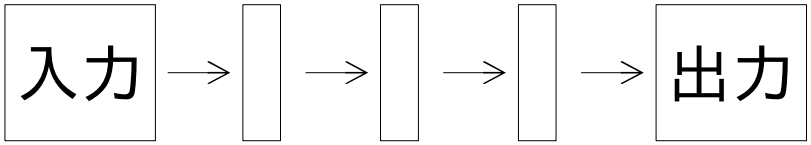
回帰の例



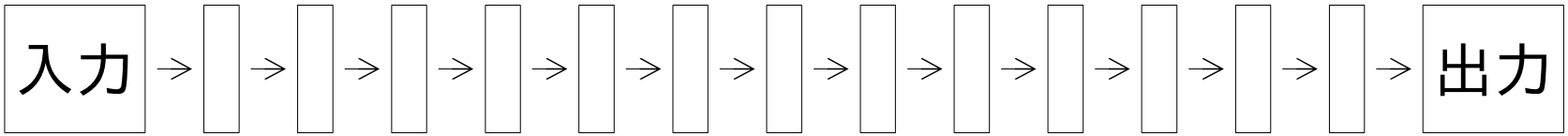
分類の例

研究背景 | 多層CNN

勾配消失問題



逆誤差伝播による学習

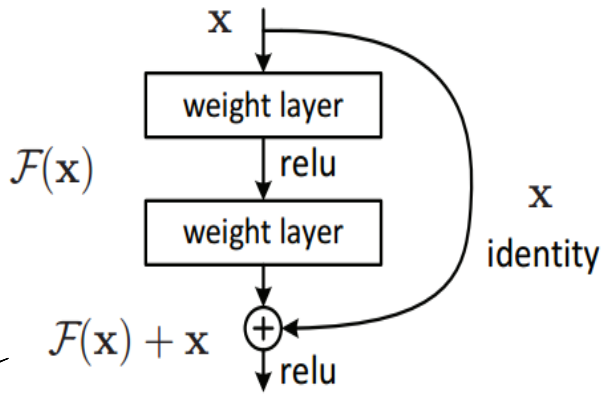


層が深くなることで、勾配情報が入力層まで届かず、学習が進まない

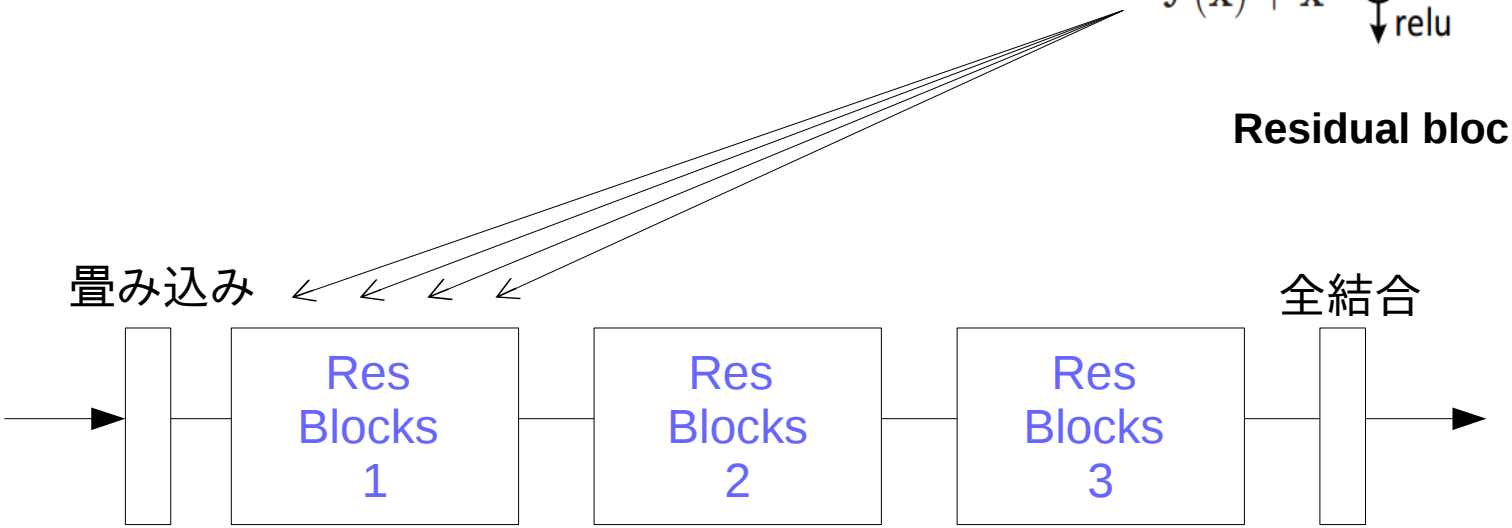
研究背景 | 多層CNN

ResNet [He+, 2015]

- skip connection によって浅い層と深い層を連結、勾配情報を浅い層にまで伝播
- 勾配消失問題の克服



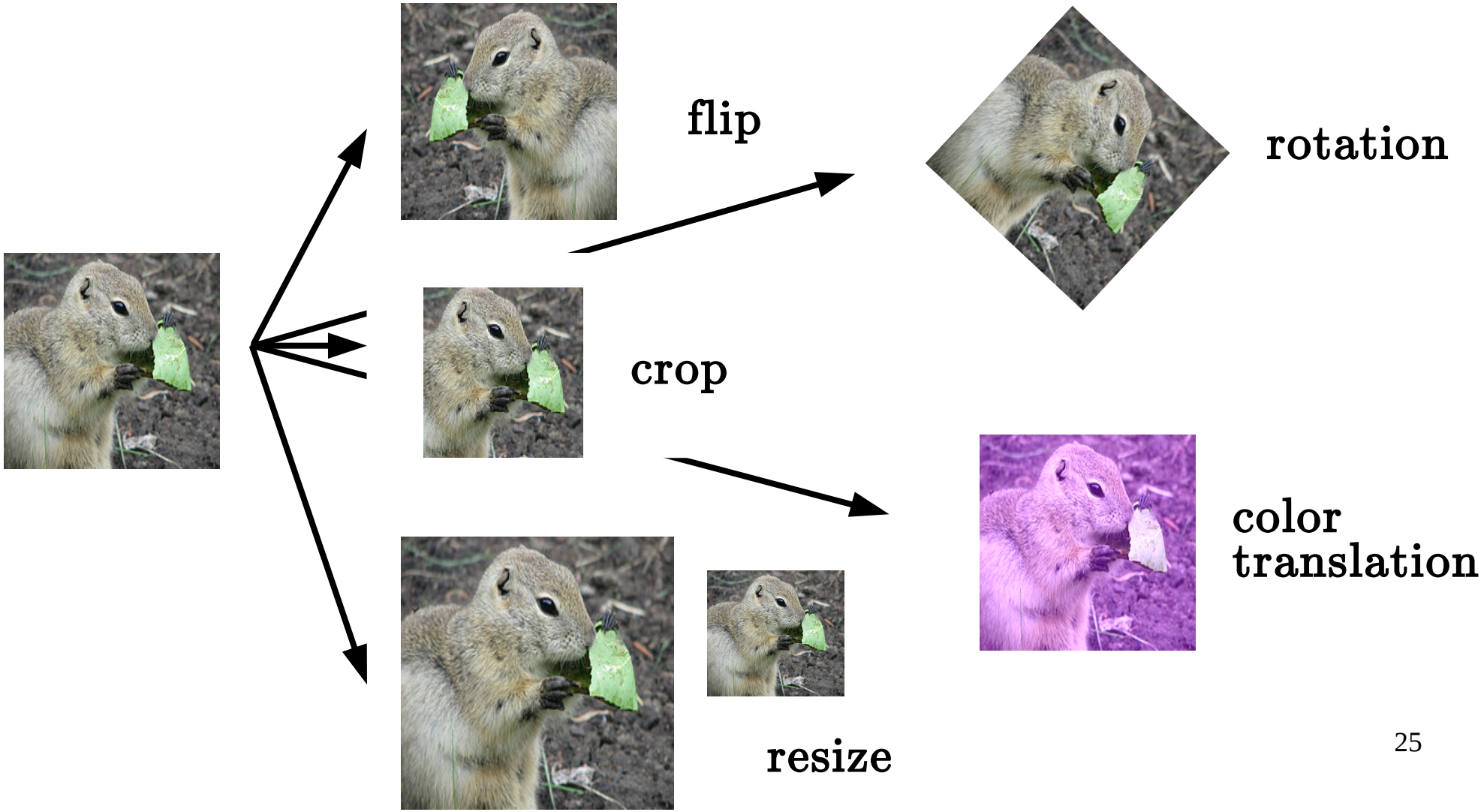
Residual block



ResNet全体図

先行研究 | data augmentation

スタンダードなdata augmentationの例



先行研究 | data augmentation

[直近の新しいdata augmentation]

masking out

- **Cutout** [Devries+, arXiv, 2017]
  - 画像の矩形領域のマスク
  - **モデルが学習する部位を毎回ランダム に変えることで、擬似的にデータを多様にする**

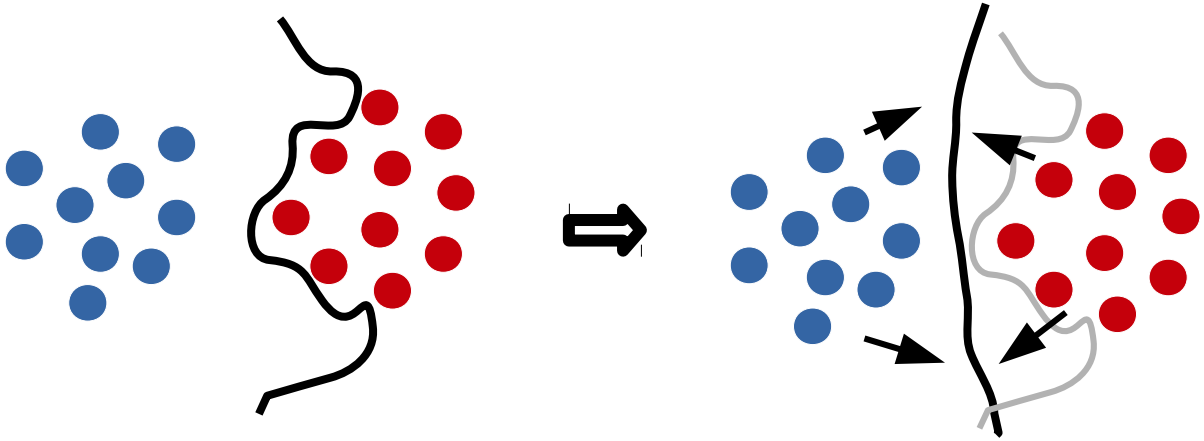
[欠点]

- マスクした部分が学習の際に無駄に

先行研究 | data augmentation

[直近の新しいdata augmentation]

- **Mixup** [Zhang+, ICLR, 2018]
  - 2枚の画像のアルファブレンド
  - クラス間の決定境界に摂動を加え、
    過学習したクラス分類を防ぐ

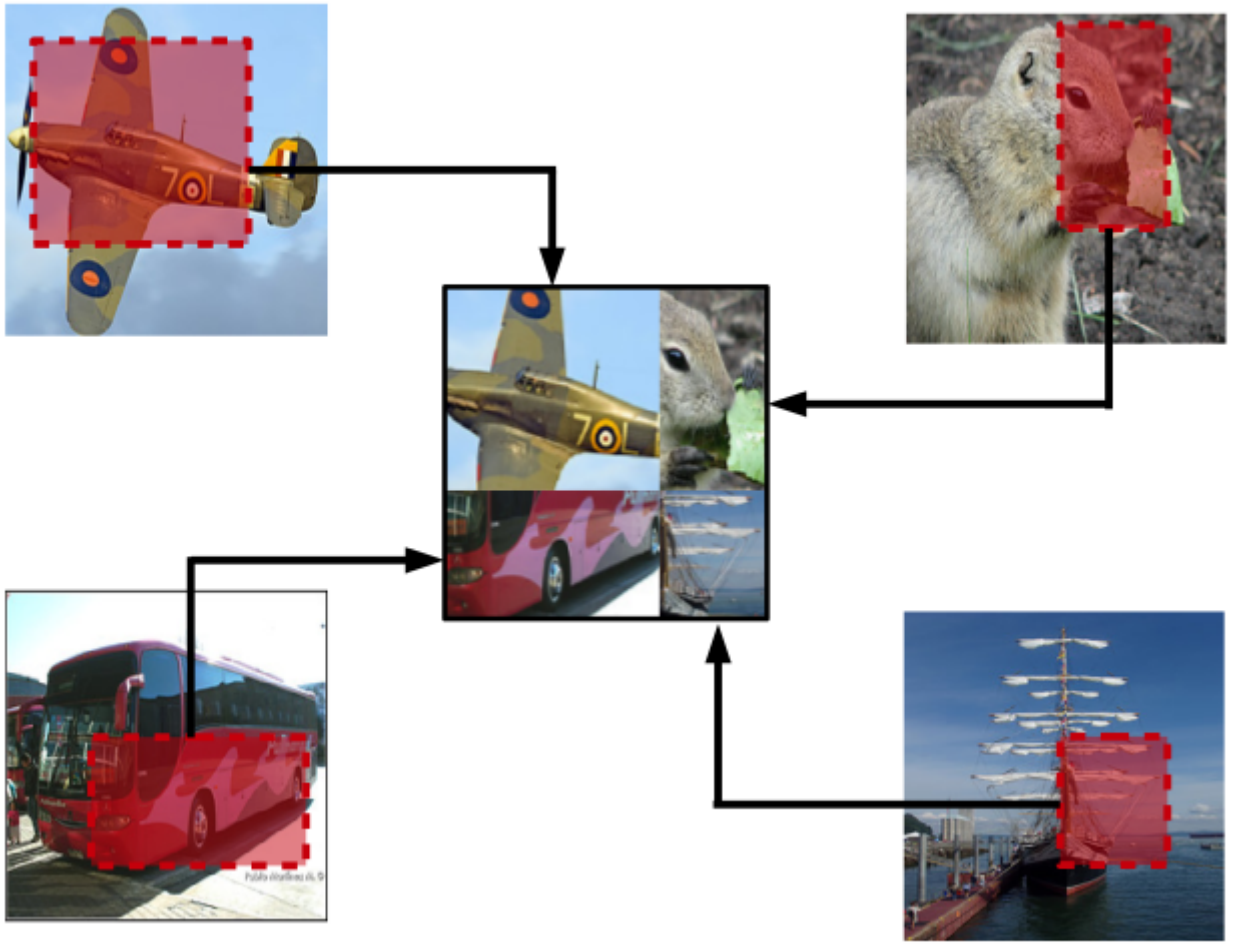


[欠点]

- データセットにない特徴を作り出し、意図しない学習が起こりうる

提案手法

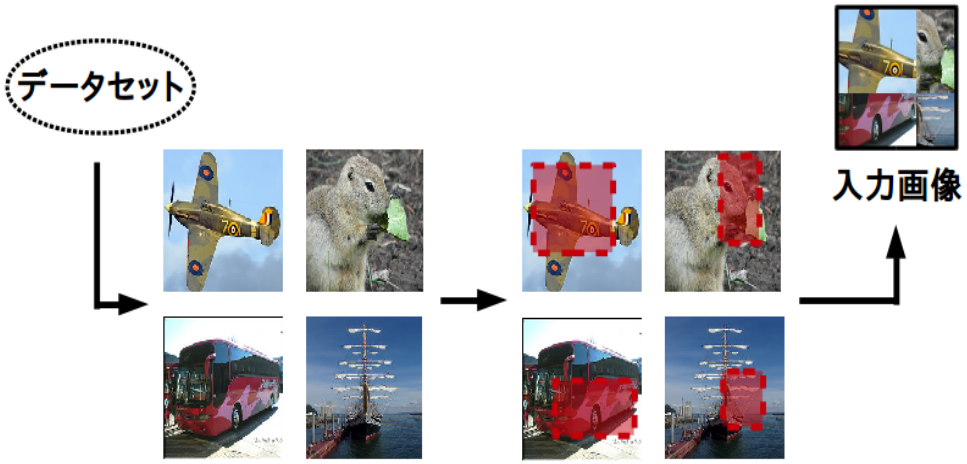
コンセプト図



提案手法

ポイント

- 画像の一部くり抜き
  - 学習毎に異なる箇所をくり抜くことで、モデルが注目し、学習する特徴を毎回変えることが出来る
  - cutoutによる過学習抑制と同じ原理
- 貼り付け
  - 4枚を1度に学習することで、**mixupによる敵対的摂動**と同じ原理
  - クラスラベルのmixによって、**クラスラベル平滑化**の恩恵を受けられる



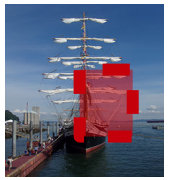
提案手法

[RICAP : random image cropping and patching]

・Cutoutに対して

[欠点] マスクした部分が学習の際に無駄に


⇨ マスクではなくくり抜きを行う（ただし、特徴が減る）

・Mixupに対して

[欠点] データセットにない特徴を作り出し、意図しない学習が起こりうる

⇨ αブレンドではなく、縦横方向に並べて貼り付ける

＋

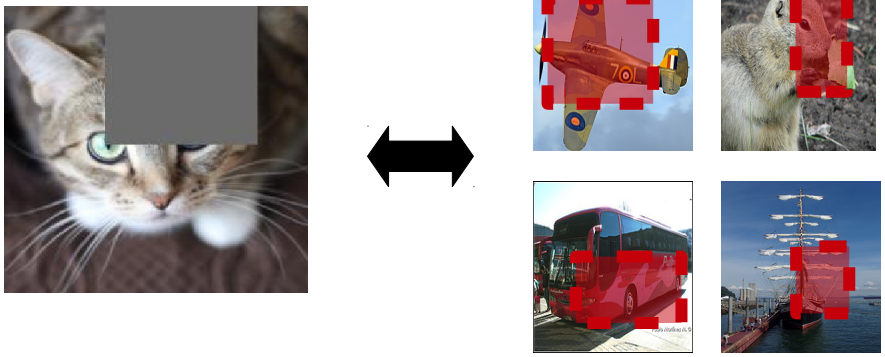
4枚の画像を使う(cutoutの特徴が減る問題を補う）

それぞれの欠点を補った上で、ハイブリッドした手法

提案手法

[RICAP : random image cropping and patching]

- Cutoutに比べて
  - マスク処理のように学習の際の無駄がない



- Mixupに比べて
  - データセットにない特徴は生まれない
  - 4枚同時に学習するため、より効率的

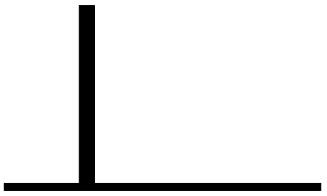
提案手法

[RICAP : random image cropping and patching]

学習方法

従来

→ CNN →



[0, 0, 1, 0 ,0, 0, ・・・]

one-hotの出力を学習

RICAP

→ CNN →

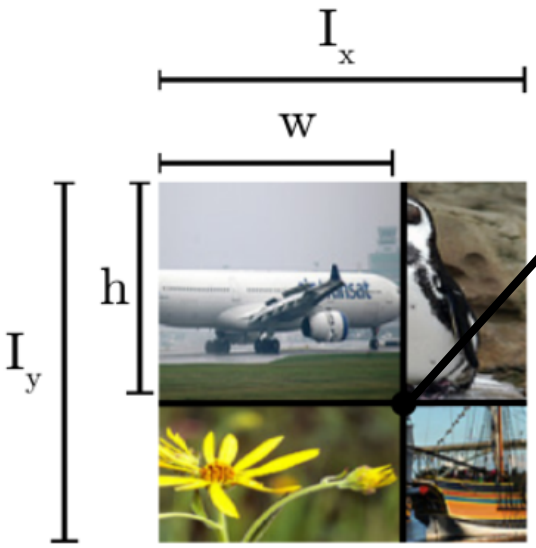
[0, 0, 0.45, 0.2, 0, 0.15, 0.2, ・・・]

0.45 x [0, 0, 1, 0 ,0, 0, ・・・]
+ 0.2 x [0, 0, 0, 1, 0, 0, ・・・]
+ 0.15 x [0, 0, 0, 0, 0, 1, 0, ・・・]
+ 0.2 x [0, 0, 0, 0, 0, 0, 1, ・・・]

各画像のラベルを
専有面積でmix

提案手法

[実装]

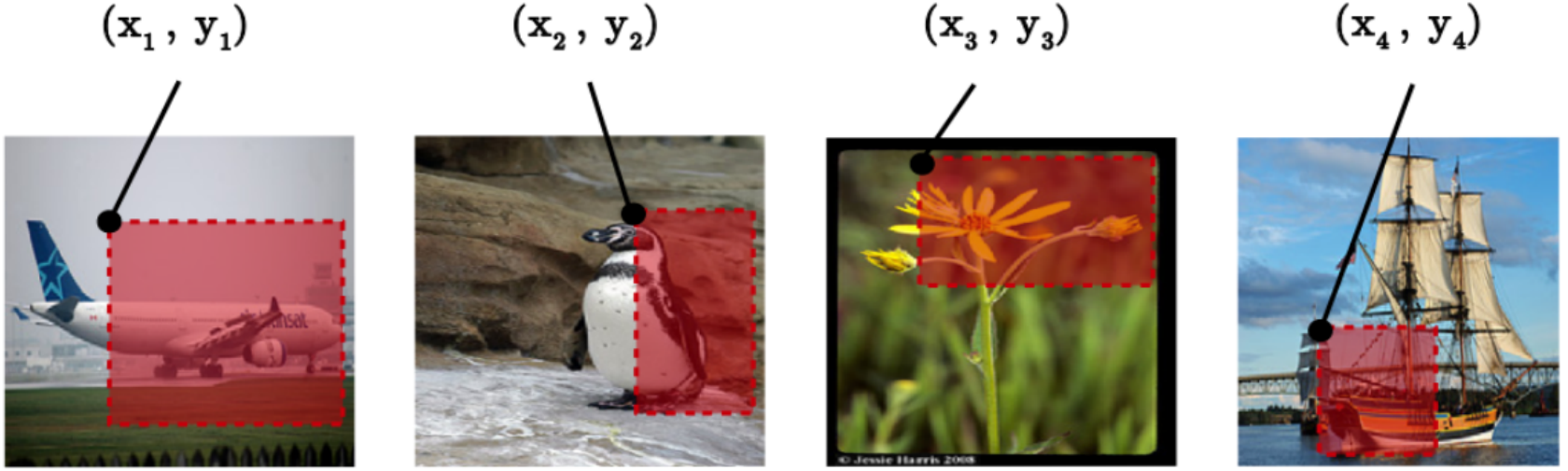


$(w,\ h)$: **boundary position**

$$\begin{cases} w' \sim \mathrm{Beta}(\beta,\beta), & h' \sim \mathrm{Beta}(\beta,\beta), \\ w = \mathrm{round}(w' I_x), & h = \mathrm{round}(h' I_y), \end{cases}$$

$$\begin{cases} x_k \sim \mathcal{U}(0, I_x - w_k), \\ y_k \sim \mathcal{U}(0, I_y - h_k). \end{cases}$$

クラスラベル：$c = \sum_{k\in\{1,2,3,4\}} W_k c_k \ \text{ for } \ W_k = \frac{w_k h_k}{I_x I_y}$

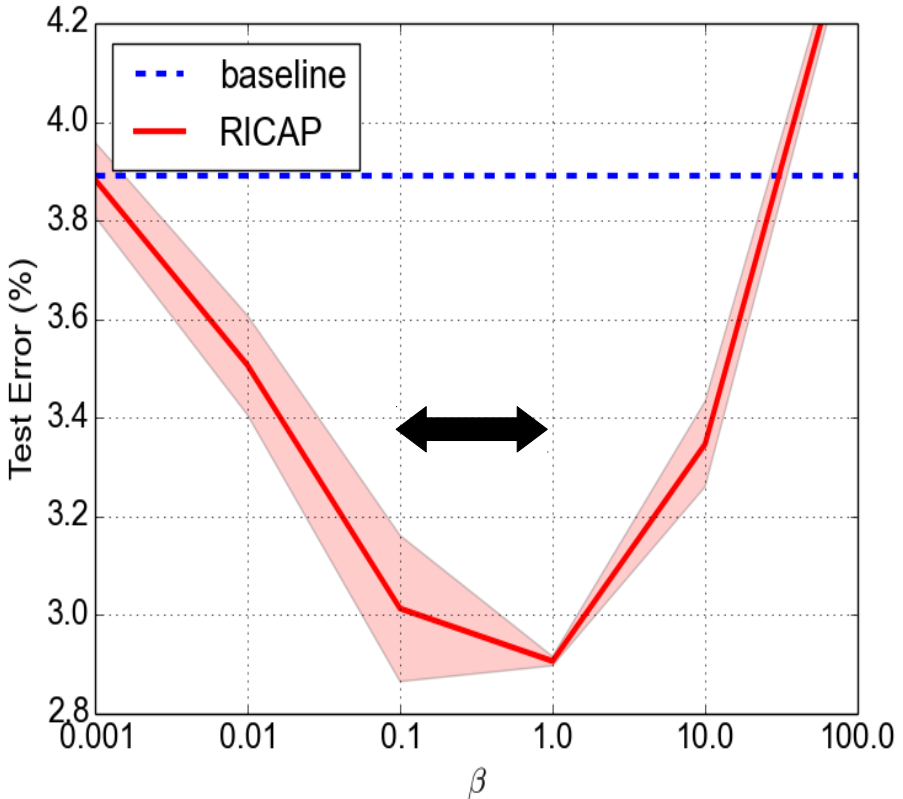


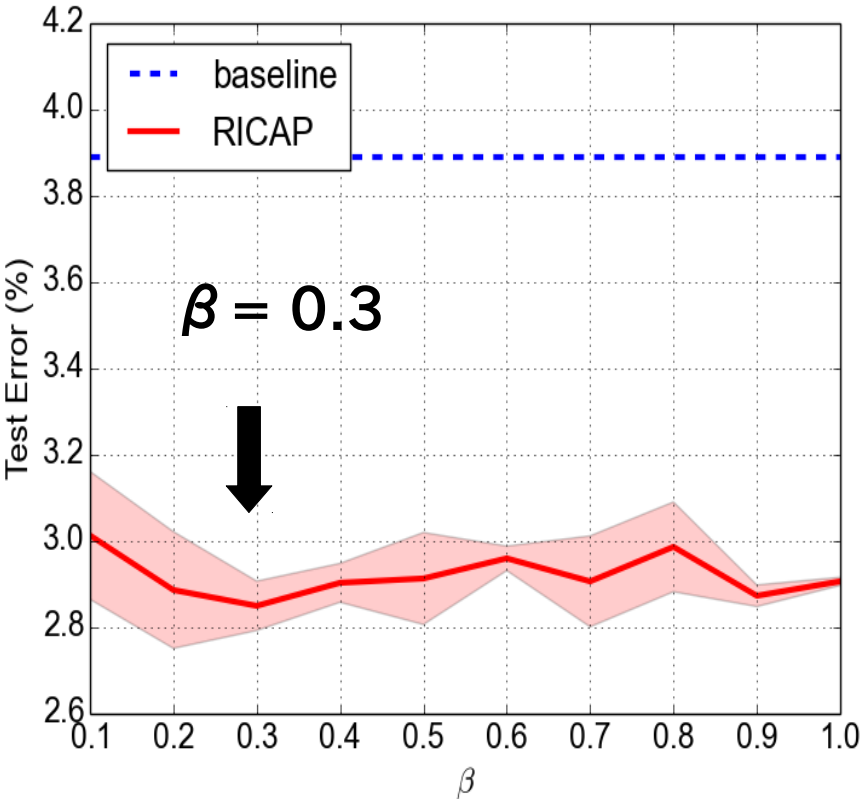
実験と結果 | 識別精度

[ベータ分布のパラメータβの探索]

- データセット : **cifar10**
- 深層CNNモデル : **Wide ResNet** [Zagoruyko+, BMVC, 2016]

$$\begin{cases} w' \sim \mathrm{Beta}(\beta, \beta), & h' \sim \mathrm{Beta}(\beta, \beta), \\ w = \mathrm{round}(w' I_x), & h = \mathrm{round}(h' I_y), \end{cases}$$

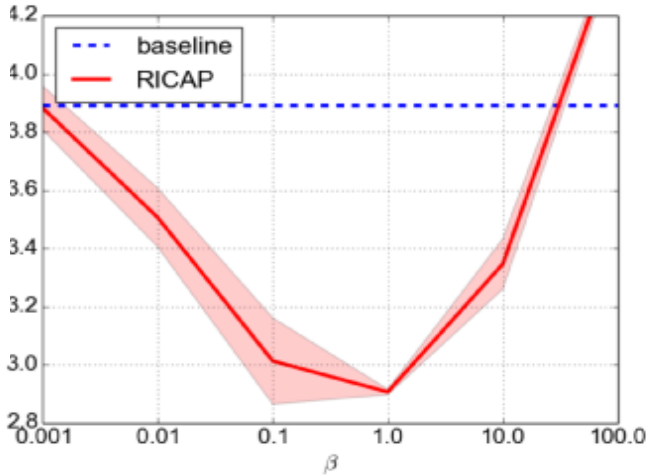




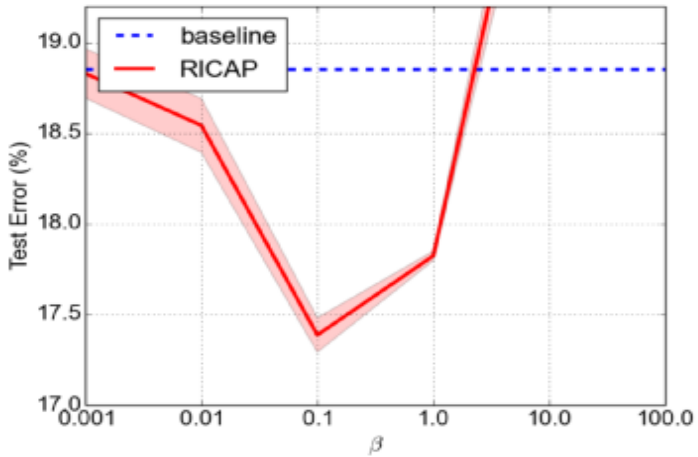
実験と結果 | 識別精度

Boundary positionの調整

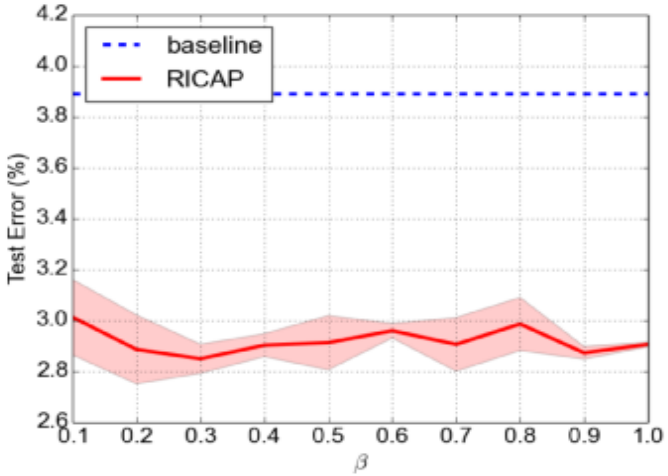
$$\frac{w}{I_x} \sim \mathrm{Beta}(\beta, \beta),$$
$$\frac{w}{I_y} \sim \mathrm{Beta}(\beta, \beta),$$
$$\beta \in (0, \infty).$$



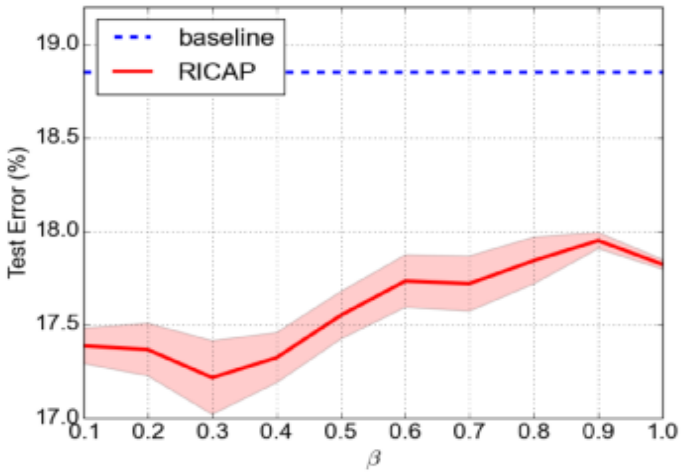
CIFAR-10



CIFAR-100



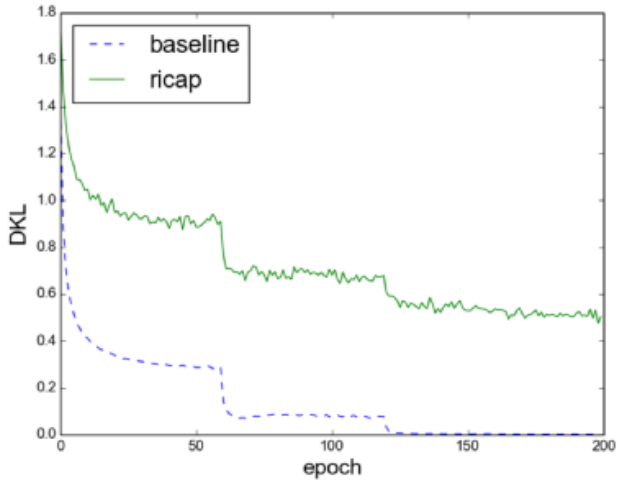
CIFAR-10



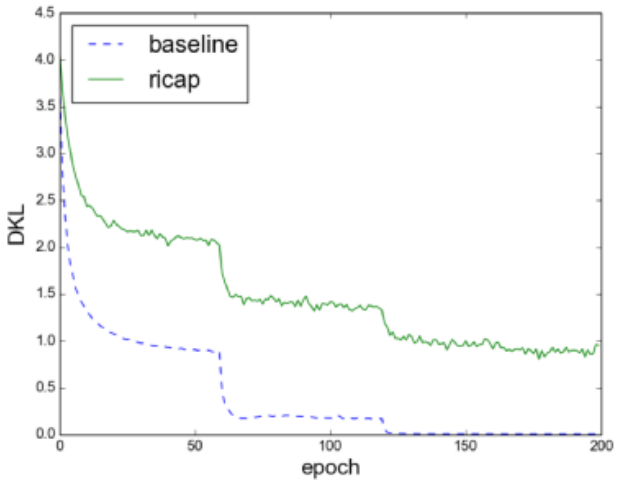
CIFAR-100

実験と結果 | Ablation Study

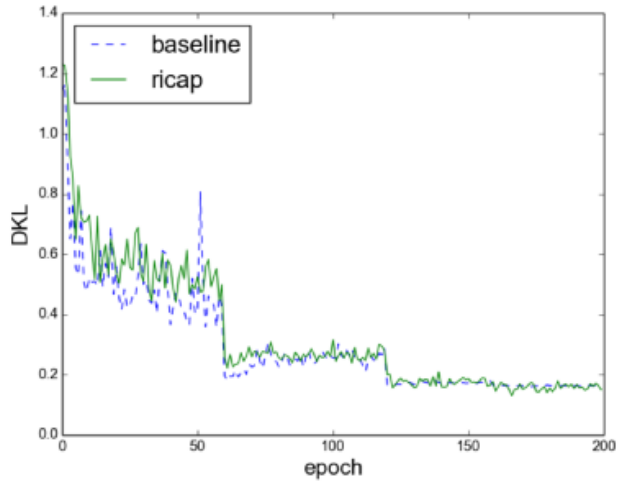
train loss, test loss



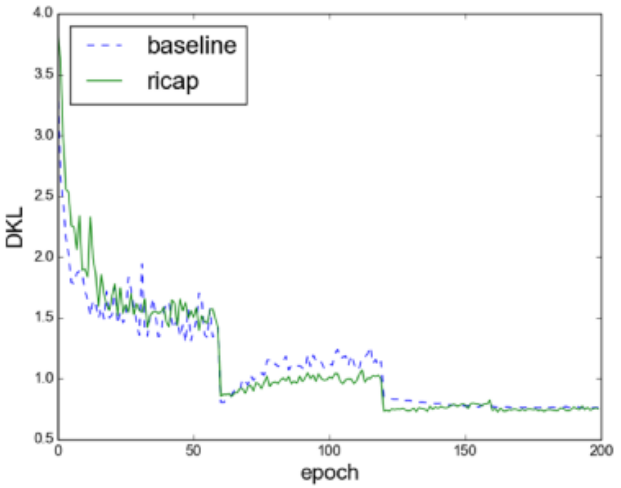
(a) train losses of WideResNet on CIFAR-10



(b) train losses of WideResNet on CIFAR-100



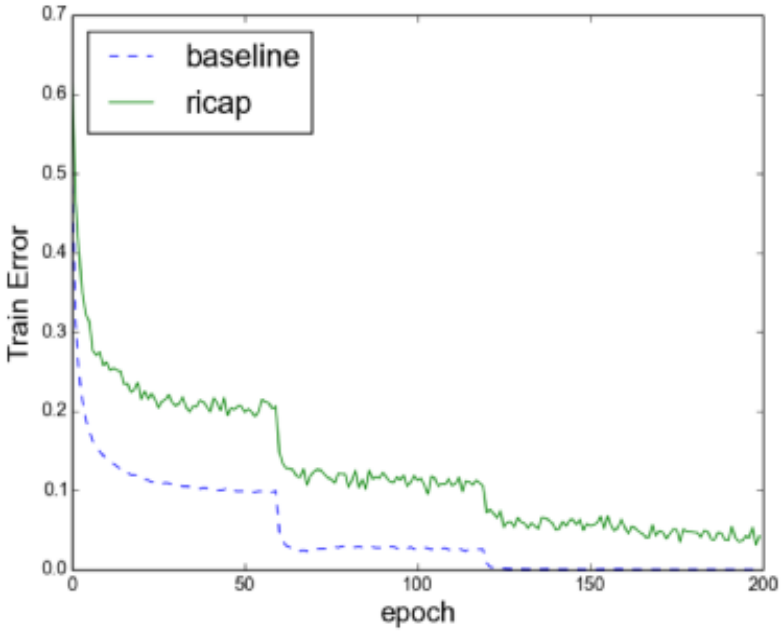
(c) test losses of WideResNet on CIFAR-10



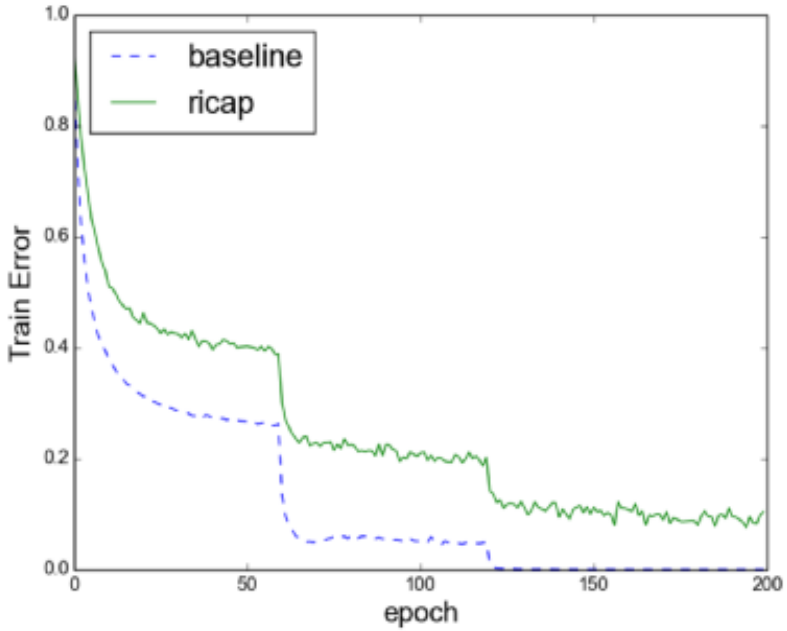
(d) test losses of WideResNet on CIFAR-100

実験と結果 | Ablation Study

train accuracy



(e) train errors of WideResNet on CIFAR-10



(f) train errors of WideResNet on CIFAR-100

実験と結果 | 他タスクへの応用

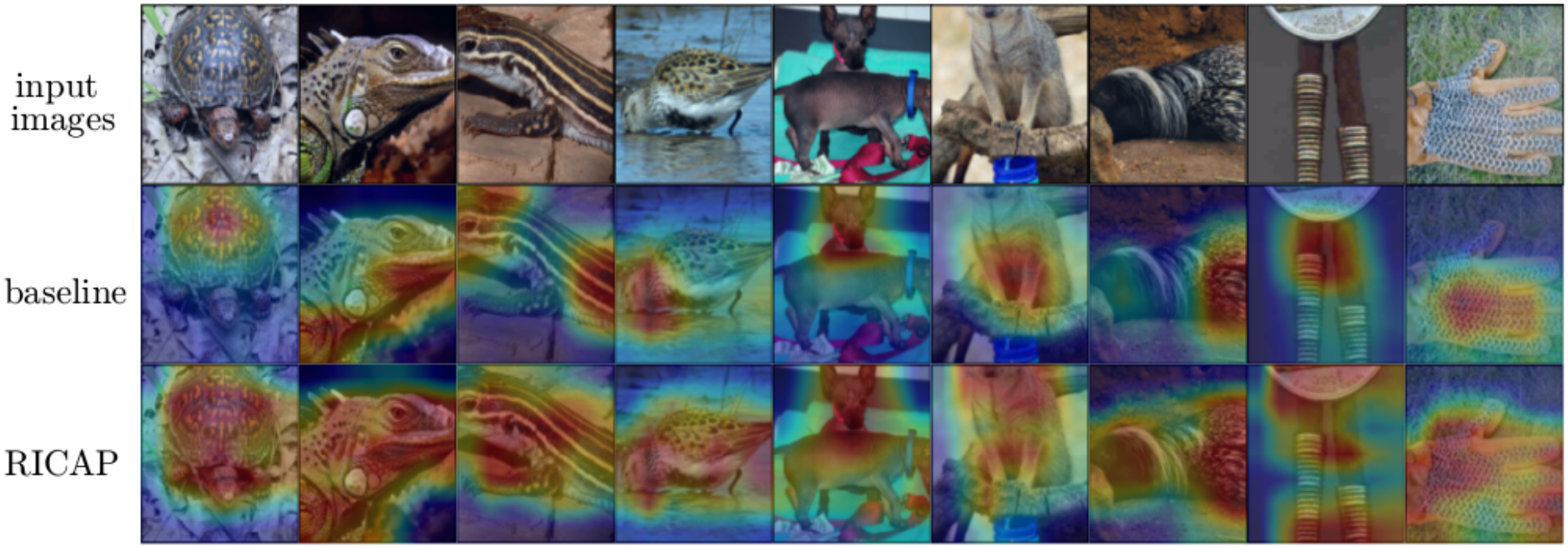
Image-caption retrieval タスクへの適用

↳ 画像 <=> キャプション　の相互検索タスク

Model	Caption Retrieval				Image Retrieval			
	R@1	R@5	R@10	Med r	R@1	R@5	R@10	Med r
Baseline	64.6	90.0	95.7	1.0	52.0	84.3	92.0	1.0
+ RICAP ($\beta = 0.3$)	65.8	90.2	96.2	1.0	52.3	84.4	92.4	1.0

実験と結果 | 可視化

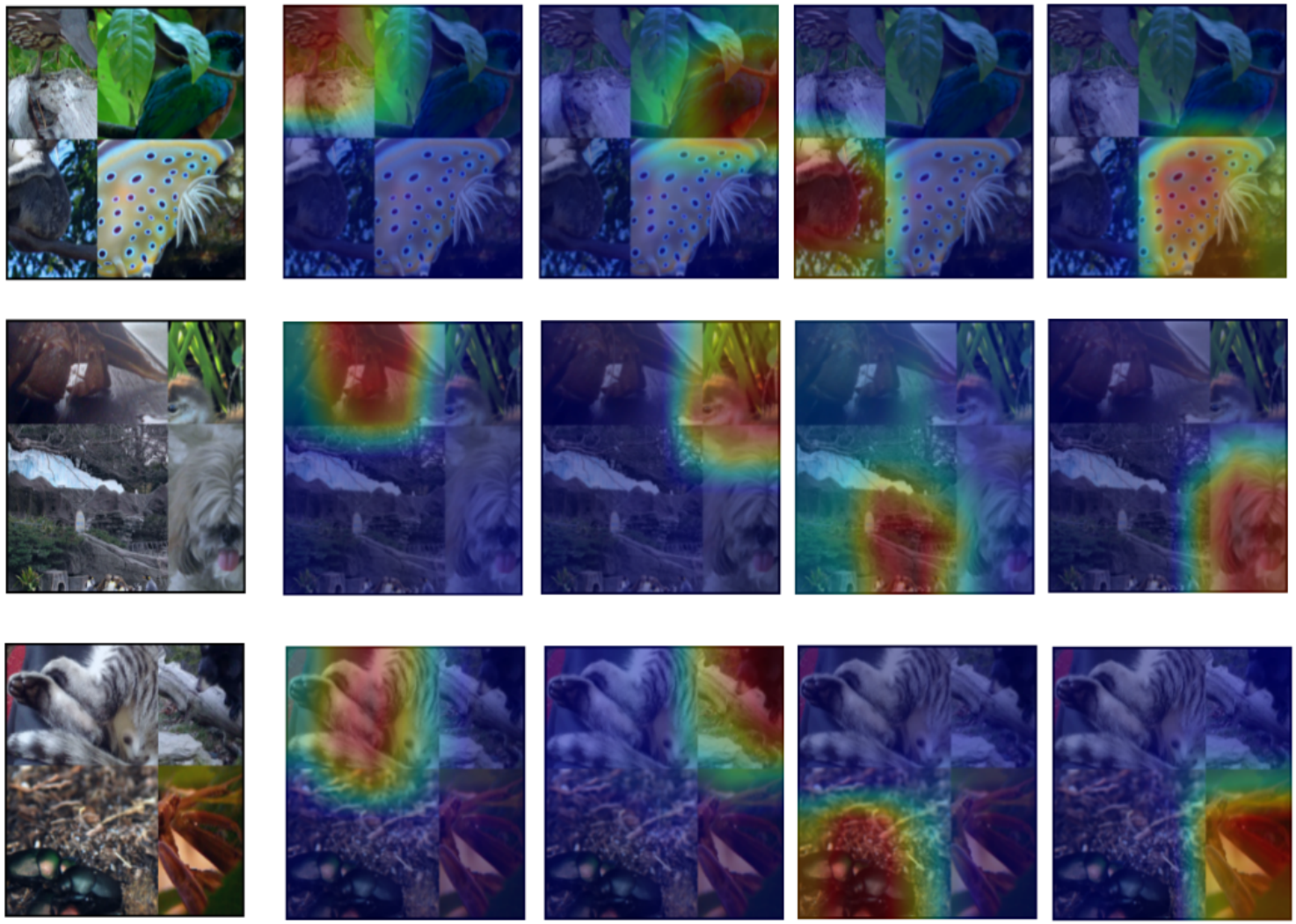
CNNの注目箇所の可視化 : CAM (Zhou, cvpr, 2016)



RICAPを用いることで、特定箇所の特徴に引っ張られず、
より詳細な特徴を捉えることが可能に

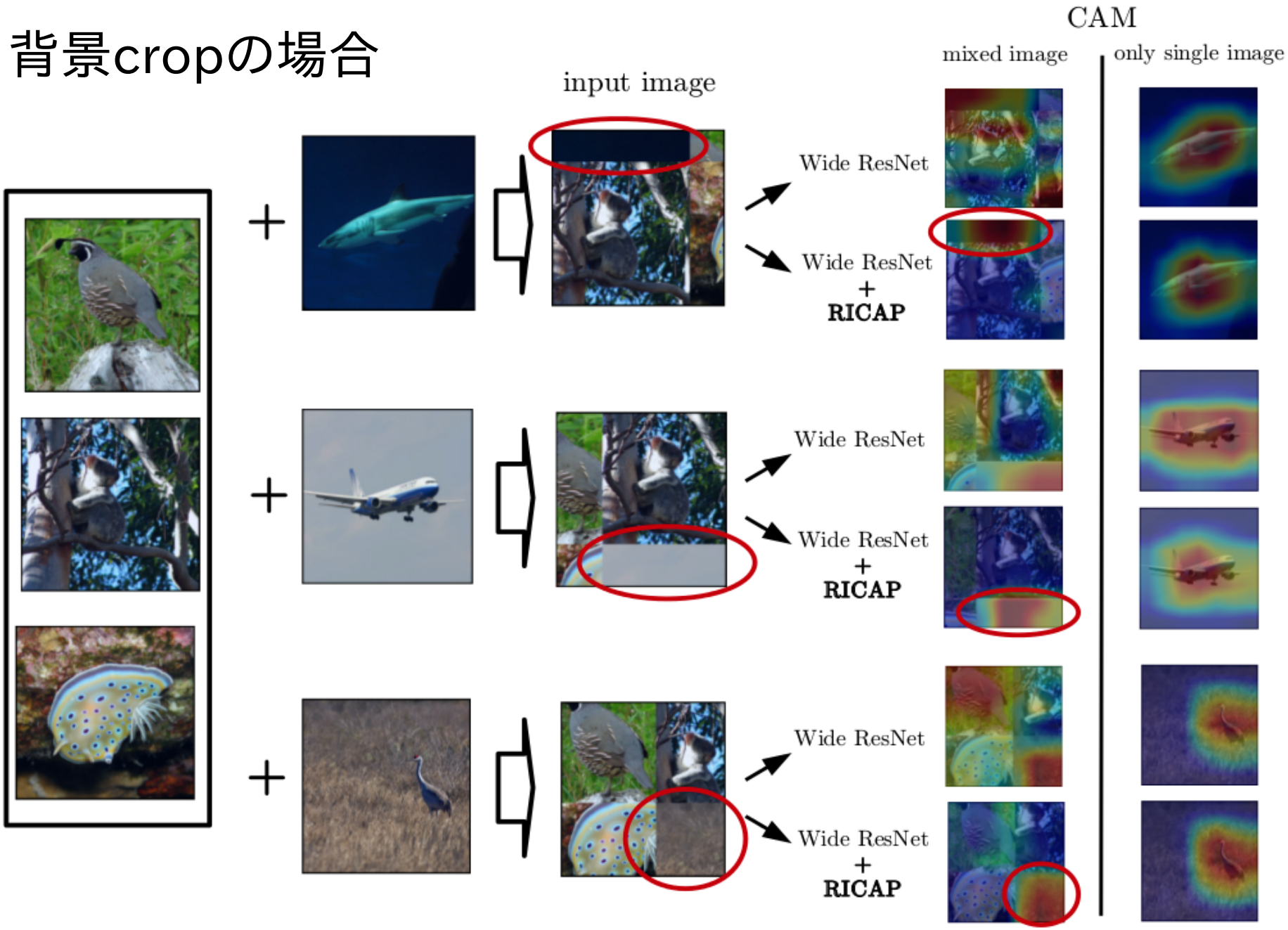
実験と結果 | 可視化

CNNの注目箇所の可視化 : CAM (Zhou, cvpr, 2016)



実験と結果 | 可視化

背景cropの場合



実験と結果 | Ablation Study

Ablation Study

Method	CIFAR-10	CIFAR-100
Baseline	3.89	18.85
+ label smoothing only ($\beta = 0.1$)	69.28	-
+ label smoothing only ($\beta = 0.3$)	62.84	-
+ label smoothing only ($\beta = 1.0$)	68.91	-
+ image mix only ($\beta = 0.1$)	3.34 $\pm$0.09	17.87 $\pm$0.22
+ image mix only ($\beta = 0.3$)	3.33 $\pm$0.10	17.95 $\pm$0.13
+ image mix only ($\beta = 1.0$)	3.70 $\pm$0.07	18.90 $\pm$0.24
+ RICAP ($\beta = 0.1$)	3.01 $\pm$0.15	17.39 $\pm$0.09
+ RICAP ($\beta = 0.3$)	**2.85** $\pm$0.06	**17.22** $\pm$0.20
+ RICAP ($\beta = 1.0$)	2.91 $\pm$0.01	17.82 $\pm$0.03