Data Augmentation via Mixing Images – part 2

Dominik Lewy

Data Augmentation via Mixing Images

$$\tilde{x} = \mathbf{B} \odot x_1 + (\mathbf{1} - \mathbf{B}) \odot x_2$$

$$\tilde{y} = \lambda y_1 + (1 - \lambda) y_2$$

Initial approaches
- 2 Smart Augmentation
- 3 Pairing Samples

Canonical Paper
- 1 Mixup

Mixup improvements
- 5 Manifold Mixup
- 4 AdaMixup

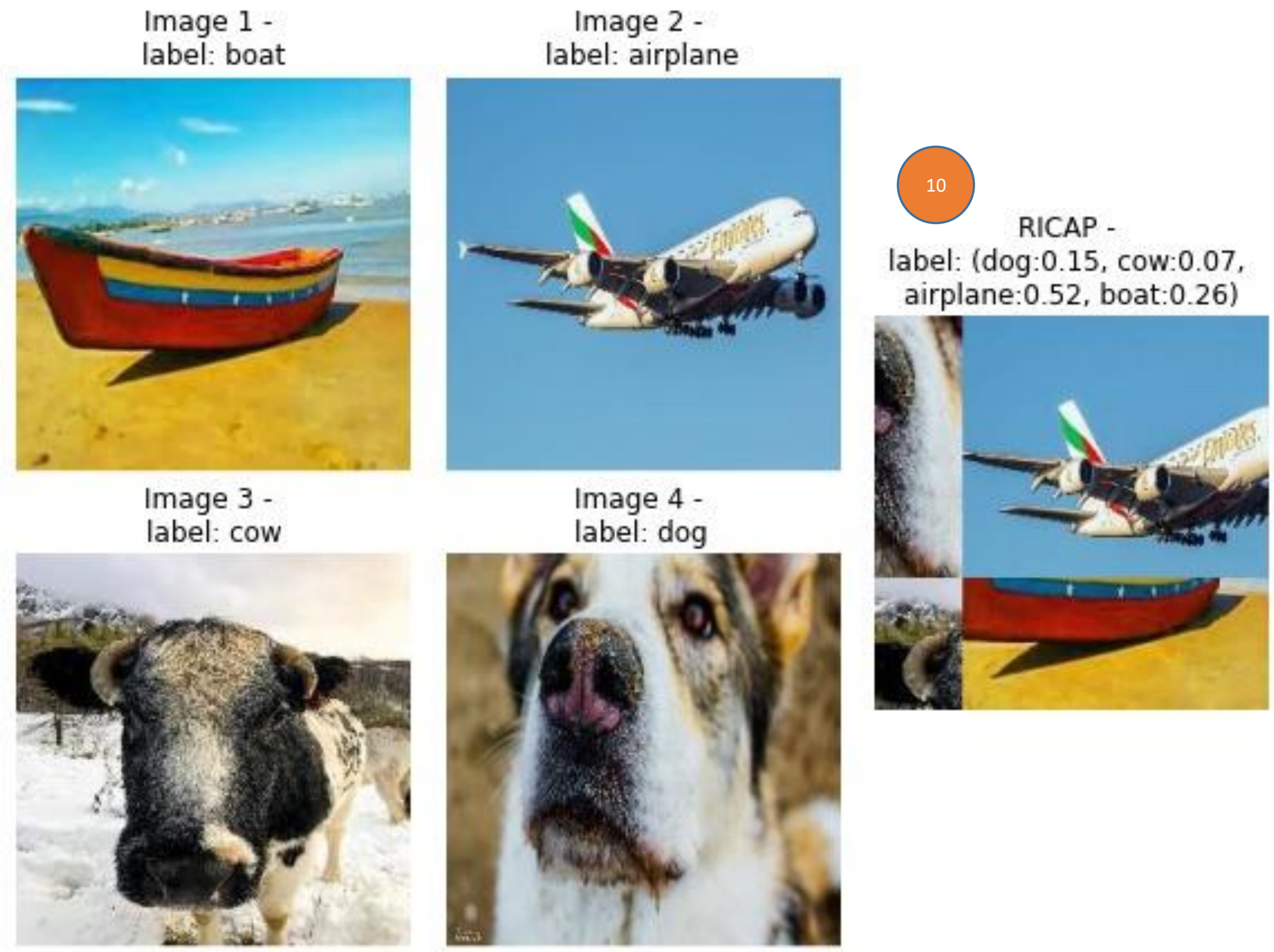
Questioning linear mixing
- 6 Mixed Examples
- 7 Cutmix
- 10 RICAP

Cutmix improvements
- 8 SmoothMix
- 9 Attentive CutMix

Data Augmentation via Mixing Images

Data Augmentation via Mixing Images

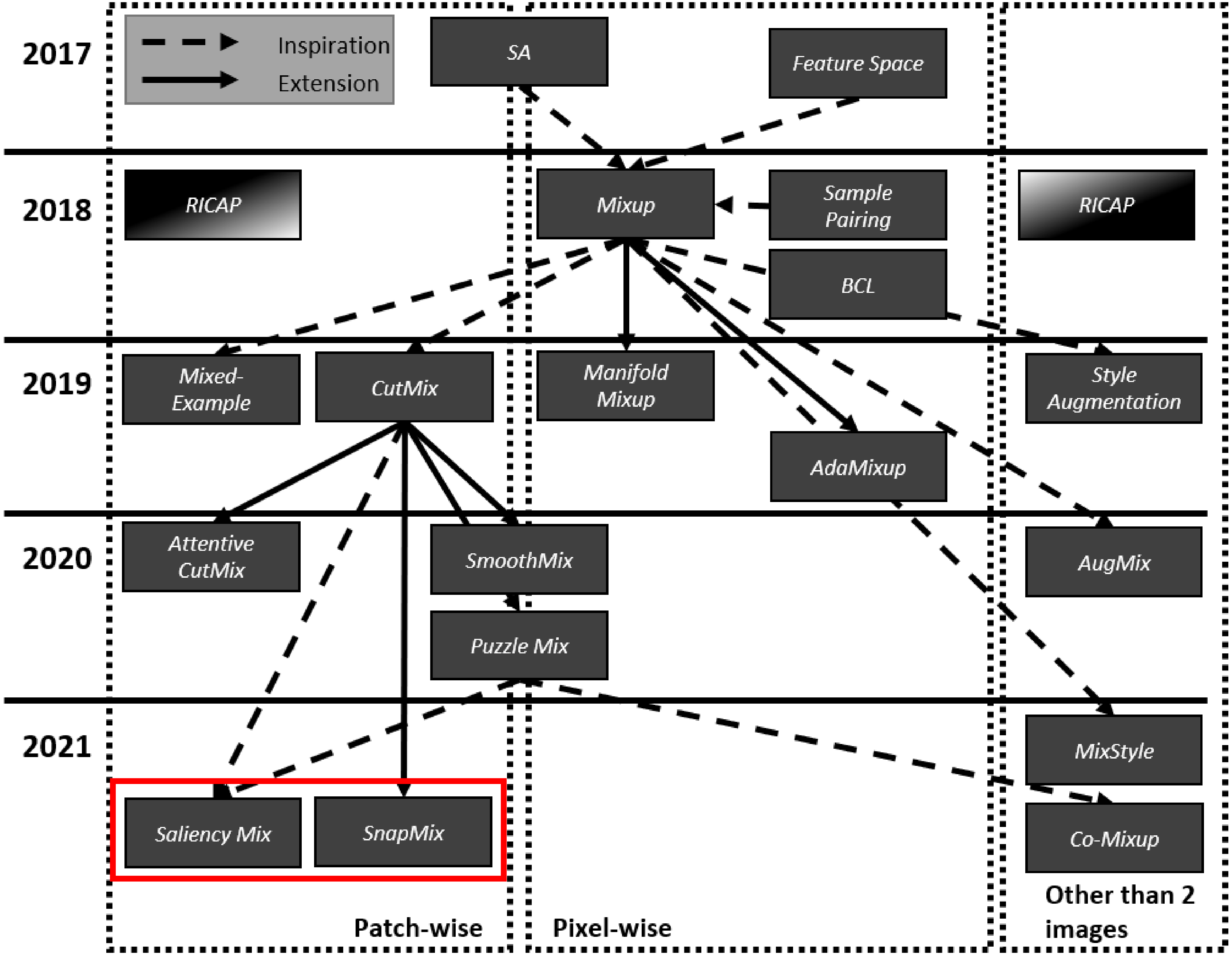
Data Augmentation via Mixing Images



Data Augmentation via Mixing Images

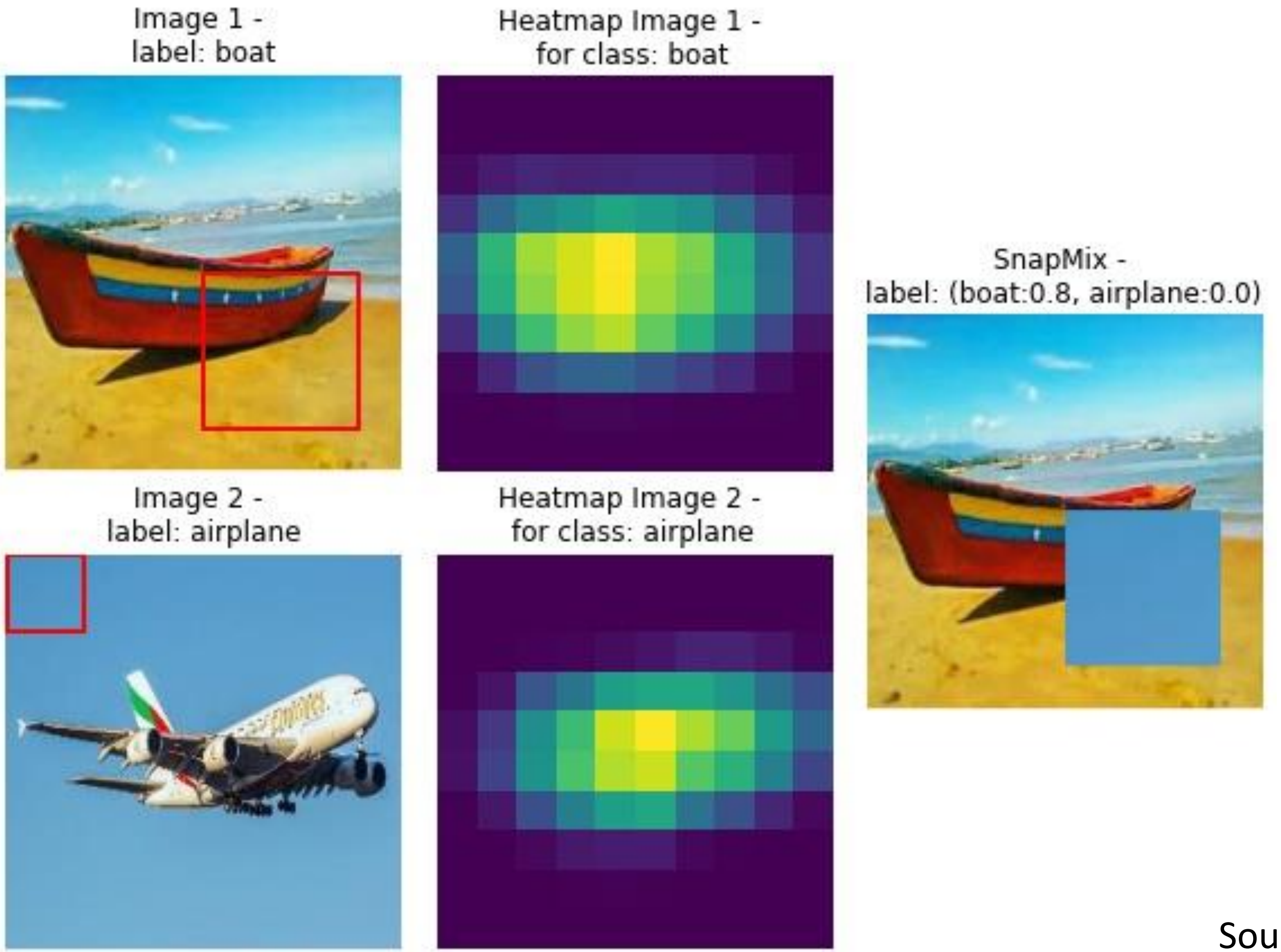
6

Data Augmentation – literature review – Dominik Lewy



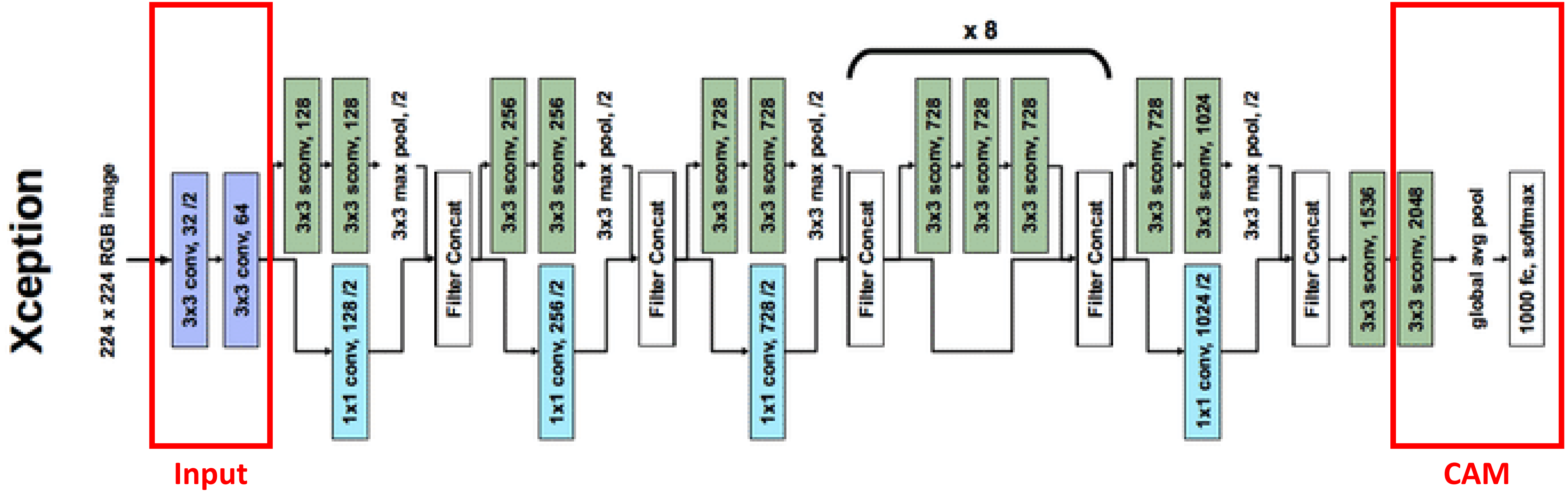
SnapMix

SNAPMIX - Shaoli Huang, Xinchao Wang, Dacheng Tao (AAAI 2021). SnapMix: Semantically Proportional Mixing for Augmenting Fine-grained Data.

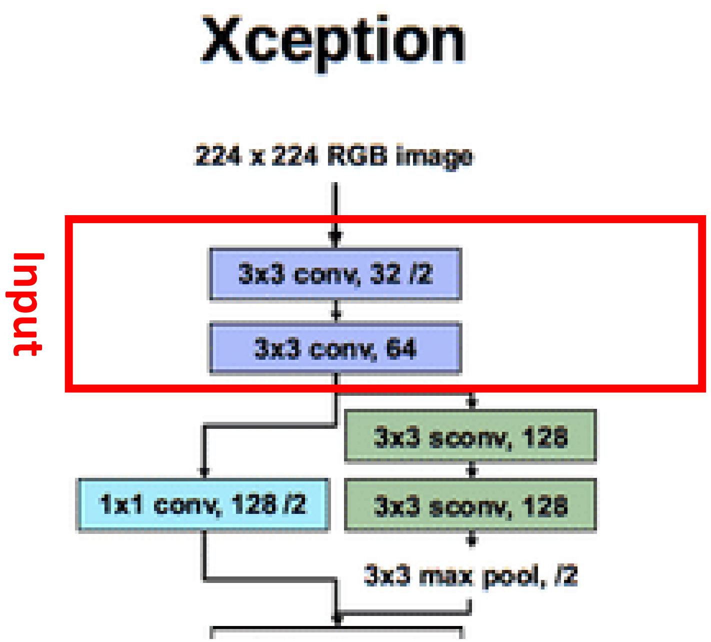


Source: own material

Class Activation Map (CAM) on Xception architecture example



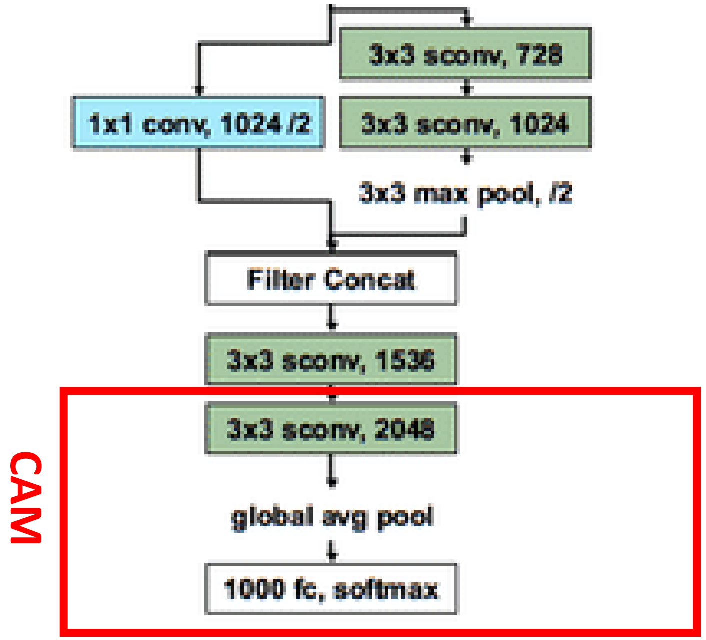
Class Activation Map (CAM) on Xception architecture example



Layer (type)	Output Shape	Param #	Connected to
input_1 (InputLayer)	[(None, 299, 299, 3)	0	
block1_conv1 (Conv2D)	(None, 149, 149, 32)	864	input_1[0][0]
block1_conv1_bn (BatchNormaliza	(None, 149, 149, 32)	128	block1_conv1[0][0]
block1_conv1_act (Activation)	(None, 149, 149, 32)	0	block1_conv1_bn[0][0]
block1_conv2 (Conv2D)	(None, 147, 147, 64)	18432	block1_conv1_act[0][0]
block1_conv2_bn (BatchNormaliza	(None, 147, 147, 64)	256	block1_conv2[0][0]
block1_conv2_act (Activation)	(None, 147, 147, 64)	0	block1_conv2_bn[0][0]
block2_sepconv1 (SeparableConv2	(None, 147, 147, 128	8768	block1_conv2_act[0][0]
block2_sepconv1_bn (BatchNormal	(None, 147, 147, 128	512	block2_sepconv1[0][0]
block2_sepconv2_act (Activation	(None, 147, 147, 128	0	block2_sepconv1_bn[0][0]

https://colab.research.google.com/drive/1wal7Qa3HA_6FBXQXjYhEpPkvwfebPogb?usp=sharing

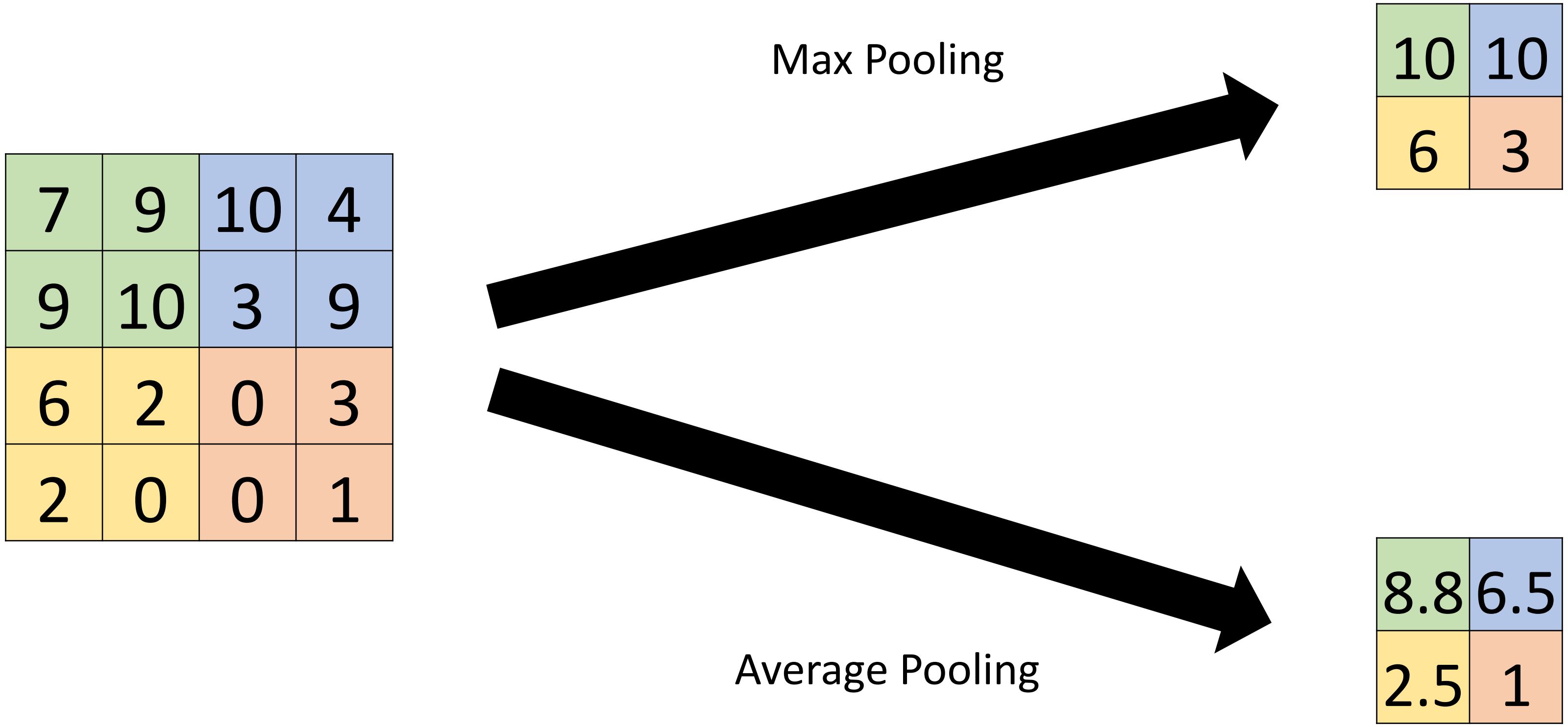
Class Activation Map (CAM) on Xception architecture example




```
________________________________________________________________________________________________
batch_normalization_3 (BatchNor (None, 10, 10, 1024) 4096        conv2d_3[0][0]
________________________________________________________________________________________________
add_11 (Add)                    (None, 10, 10, 1024) 0           block13_pool[0][0]
                                                                 batch_normalization_3[0][0]
________________________________________________________________________________________________
block14_sepconv1 (SeparableConv (None, 10, 10, 1536) 1582080     add_11[0][0]
________________________________________________________________________________________________
block14_sepconv1_bn (BatchNorma (None, 10, 10, 1536) 6144        block14_sepconv1[0][0]
________________________________________________________________________________________________
block14_sepconv1_act (Activatio (None, 10, 10, 1536) 0           block14_sepconv1_bn[0][0]
________________________________________________________________________________________________
block14_sepconv2 (SeparableConv (None, 10, 10, 2048) 3159552     block14_sepconv1_act[0][0]
________________________________________________________________________________________________
block14_sepconv2_bn (BatchNorma (None, 10, 10, 2048) 8192        block14_sepconv2[0][0]
________________________________________________________________________________________________
block14_sepconv2_act (Activatio (None, 10, 10, 2048) 0           block14_sepconv2_bn[0][0]
________________________________________________________________________________________________
avg_pool (GlobalAveragePooling2 (None, 2048)         0           block14_sepconv2_act[0][0]
________________________________________________________________________________________________
predictions (Dense)             (None, 1000)         2049000     avg_pool[0][0]
================================================================================================
```


https://colab.research.google.com/drive/1wal7Qa3HA_6FBXQXjYhEpPkvwfebPogb?usp=sharing

Class Activation Map (CAM) on Xception architecture example

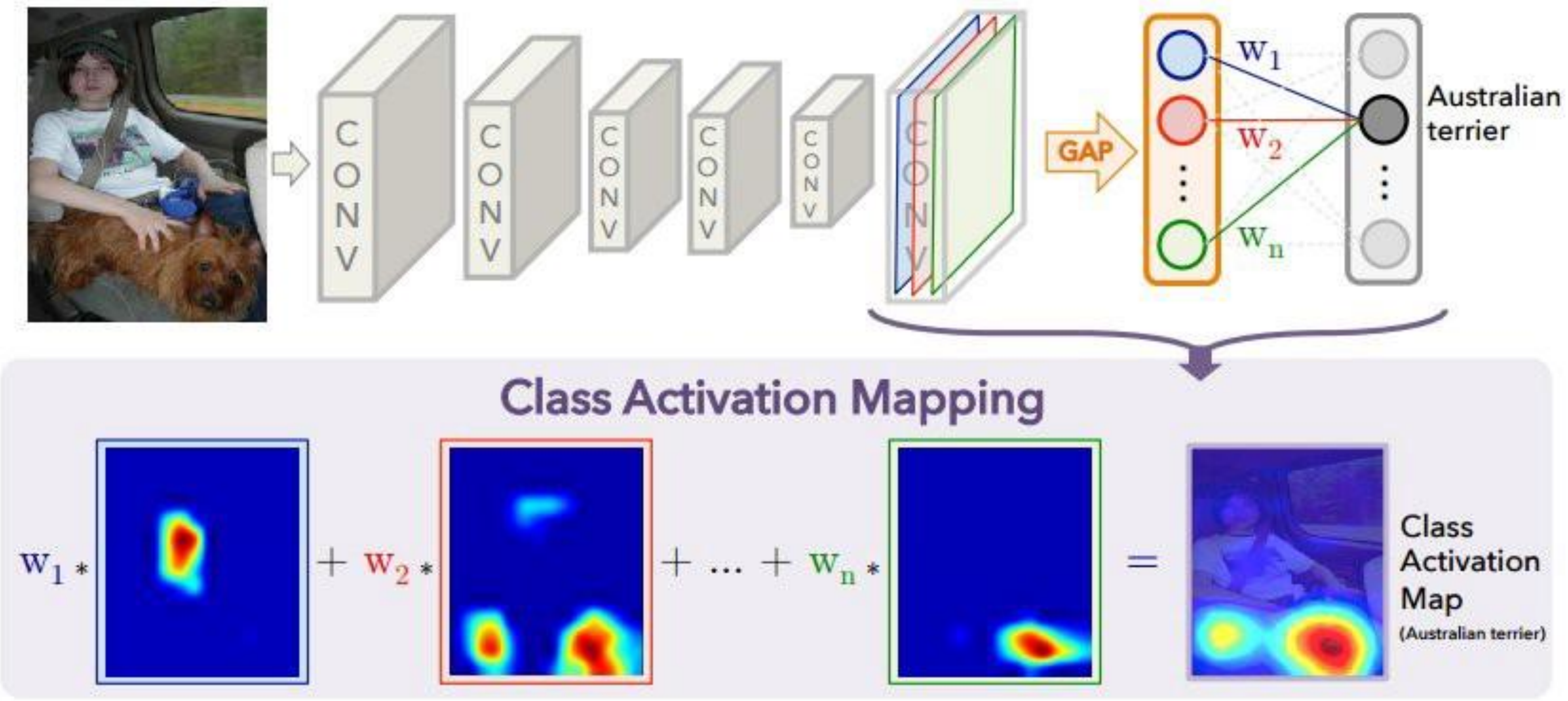


https://www.di.ens.fr/~josef/publications/Oquab15.pdf

https://arxiv.org/pdf/1512.04150.pdf

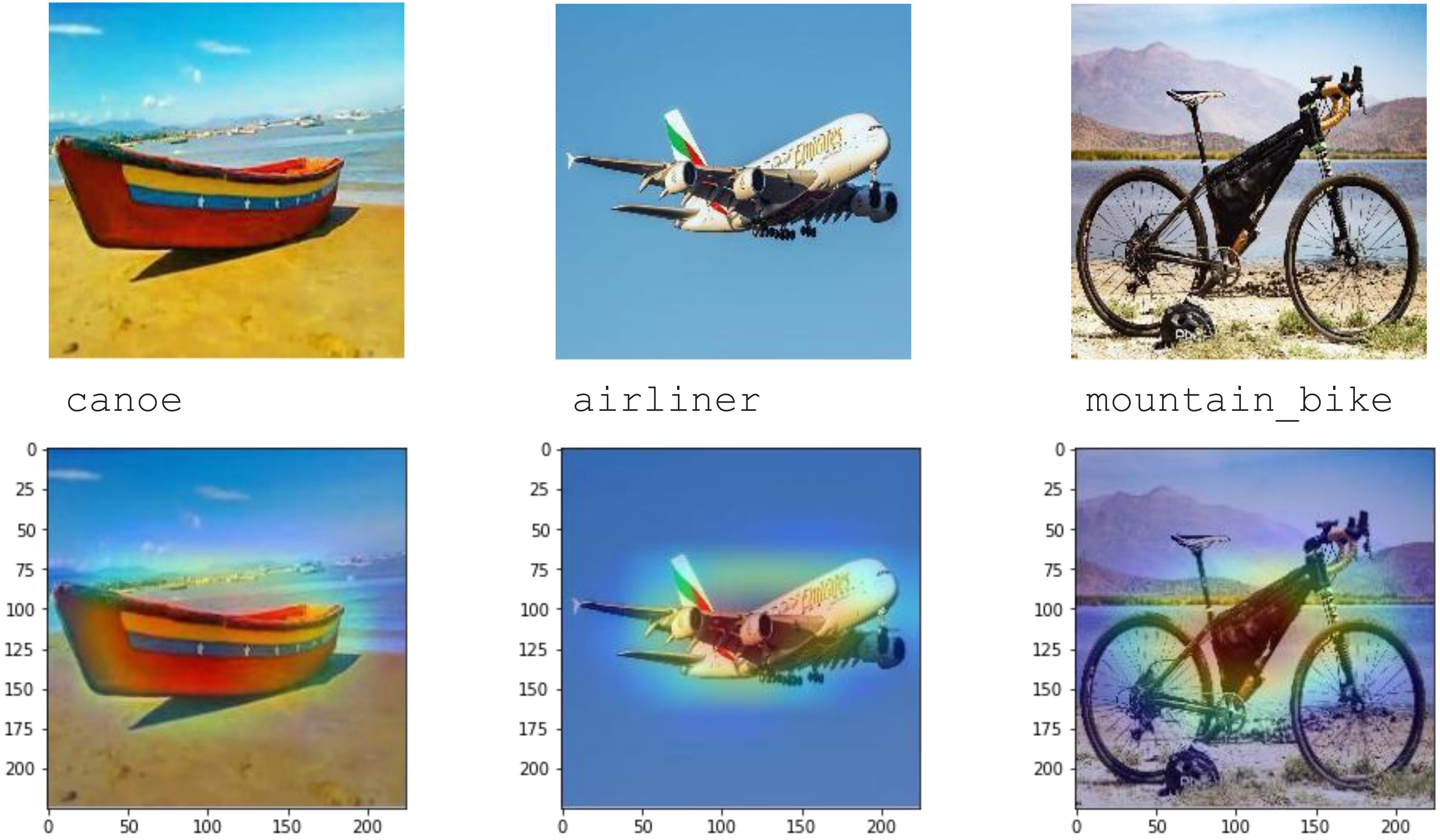
Source: own material

Class Activation Map (CAM) on Xception architecture example



Source: https://arxiv.org/pdf/1512.04150.pdf

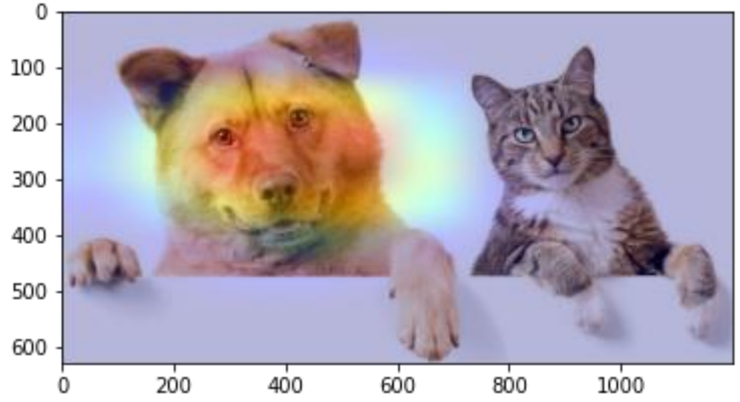
Class Activation Map (CAM) on Xception architecture example

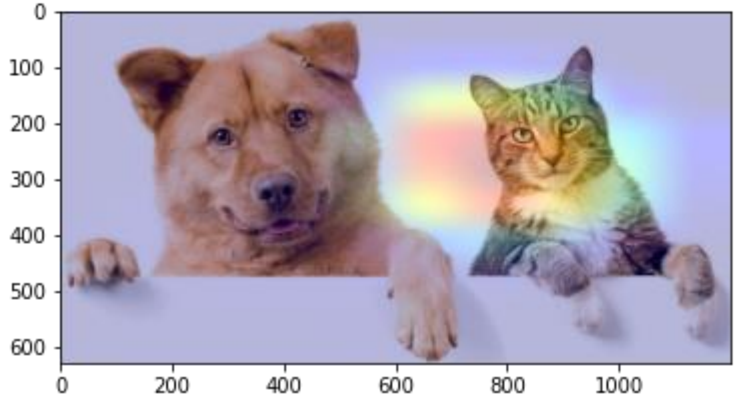


Source: own material

Class Activation Map (CAM) on Xception architecture example

Predicted: [('n02112137', 'chow', 0.06336529), ('n02124075', 'Egyptian_cat', 0.050370798)]

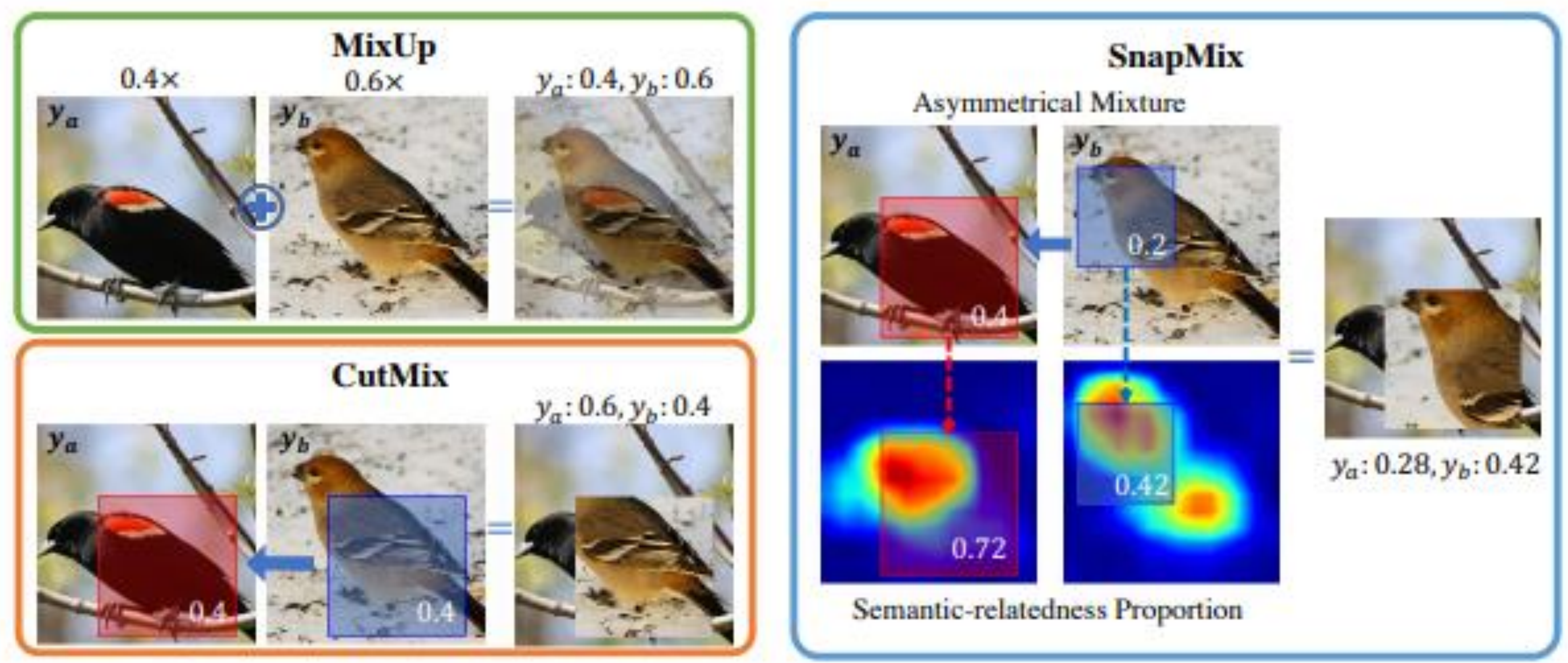




Source: own material

SnapMix

SNAPMIX - Shaoli Huang, Xinchao Wang, Dacheng Tao (AAAI 2021). SnapMix: Semantically Proportional Mixing for Augmenting Fine-grained Data.



SnapMix

SNAPMIX - Shaoli Huang, Xinchao Wang, Dacheng Tao (AAAI 2021). SnapMix: Semantically Proportional Mixing for Augmenting Fine-grained Data.

$$\tilde{x} = (I - B_{\lambda^1}) \odot x_1 + T_\theta(B_{\lambda^2} \odot x_2)$$

where B_{λ^1} and B_{λ^2} are two binary masks containing random box regions with the area ratio λ^1 and λ^2, respectively, and T_θ is a function that maps the patch from x_2 onto the patch in x_1.

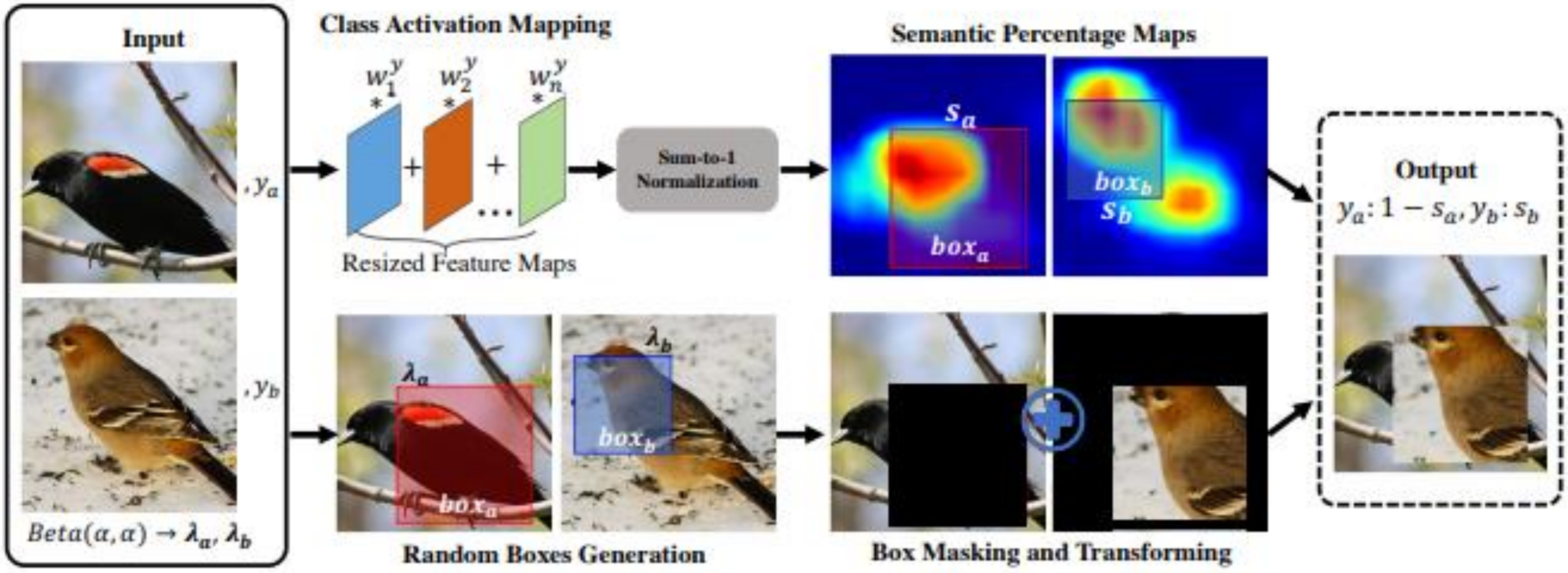
$$S(x_i) = \frac{CAM(x_i)}{sum(CAM(x_i))}$$

$$p_1 = 1 - sum(B_{\lambda^1} \odot S(x_1)), \qquad p_2 = sum(B_{\lambda^2} \odot S(x_2))$$

where $p_1, p_2 \in [0,1]$ are partial labels assigned to classes corresponding to the class of images 1 and 2, respectively.

SnapMix

SNAPMIX - Shaoli Huang, Xinchao Wang, Dacheng Tao (AAAI 2021). SnapMix: Semantically Proportional Mixing for Augmenting Fine-grained Data.



SnapMix

SNAPMIX - Shaoli Huang, Xinchao Wang, Dacheng Tao (AAAI 2021). SnapMix: Semantically Proportional Mixing for Augmenting Fine-grained Data.

Table 1: Performance comparison(Mean Acc.%) of methods using backbone networks *Resnet-18* and *Resnet-34* on fine-grained datasets. Each method's improvement over the baseline is shown in the brackets.

	CUB		Cars		Aircraft	
	Res18	Res34	Res18	Res34	Res18	Res34
Baseline	82.35	84.98	91.15	92.02	87.80	89.92
CutOut	80.54 (-1.81)	83.36 (-1.62)	91.83(+0.68)	92.84 (+0.82)	88.58 (+0.78)	89.90 (-0.02)
MixUp	83.17 (+0.82)	85.22 (+0.24)	91.57 (+0.42)	93.28 (+1.26)	89.82 (+2.02)	91.02 (+1.1)
CutMix	80.16 (-2.19)	85.69 (+0.71)	92.65 (+1.50)	93.61 (+1.59)	89.44 (+1.64)	91.26 (+1.34)
SnapMix	**84.29 (+1.94)**	**87.06 (+2.08)**	**93.12(+1.97)**	**93.95 (+1.93)**	**90.17 (+2.37)**	**92.36 (+2.44)**

Table 2: Performance comparison(Mean Acc.%) of methods using backbone networks *Resnet-50* and *Resnet-101* on fine-grained datasets. Each method's improvement over the baseline is shown in the brackets.

	CUB		Cars		Aircraft	
	Res50	Res101	Res50	Res101	Res50	Res101
Baseline	85.49	85.62	93.04	93.09	91.07	91.59
CutOut	83.55 (-1.94)	84.70 (-0.92)	93.76 (+0.72)	94.16 (+1.07)	91.23 (+0.16)	91.79 (+0.2)
MixUp	86.23 (+0.74)	87.72 (+2.1)	93.96 (+0.92)	94.22 (+1.13)	92.24 (+1.17)	92.89 (+1.3)
CutMix	86.15 (+0.66)	87.92 (+2.3)	94.18 (+1.14)	94.27 (+1.18)	92.23 (+1.16)	92.29 (+0.7)
SnapMix	**87.75 (+2.26)**	**88.45 (+2.83)**	**94.30 (+1.21)**	**94.44 (+1.35)**	**92.80 (+1.73)**	**93.74 (+2.15)**

SnapMix

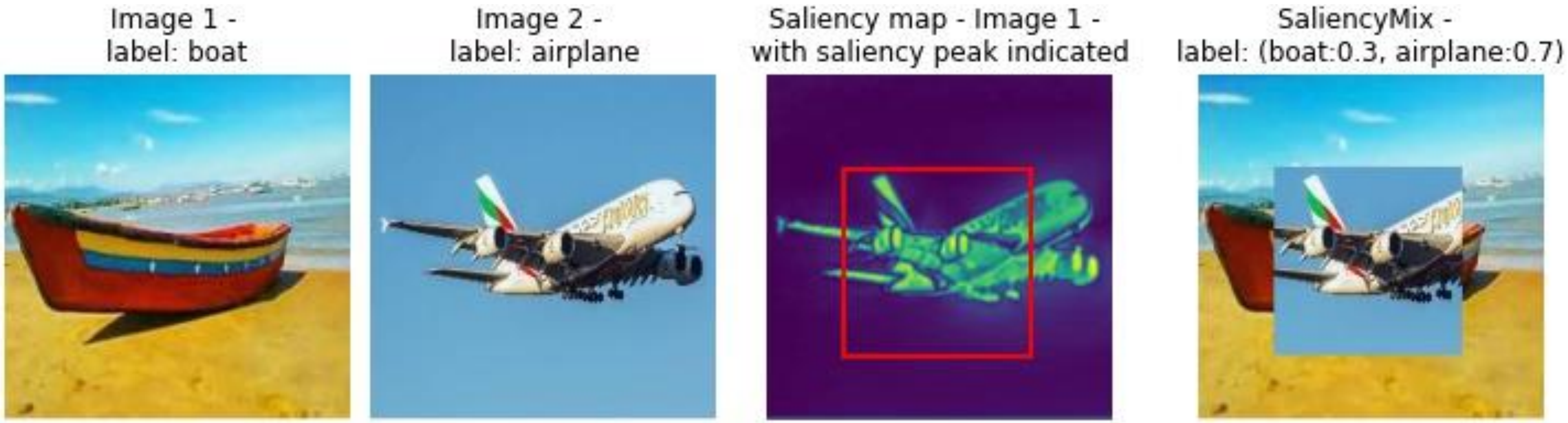
SNAPMIX - Shaoli Huang, Xinchao Wang, Dacheng Tao (AAAI 2021). SnapMix: Semantically Proportional Mixing for Augmenting Fine-grained Data.

Key points:

- The method is **dedicated to fine-grained image classification problem**, in which classes differentiate by details only
- Modification of CutMix in the two following aspects:
  - the way of label vector calculation for the mixed sample
  - **independent selection of the size and placement of the patch** in each of the two input images (the patch is not automatically pasted in the corresponding location of the target image).
- In order to calculate the label for the augmented image, the class activation map (CAM) is used.
  - CAM is a transformation applied on top of the last convolutional layer of the network so as to point locations of the class-discriminative features.
- The output of CAM is normalized to get Semantic Percent Map (SPM)
- An interesting aspect of the method is the **lack of a constraint that partial labels should sum up to 1**. This way it is possible to indicate in the label that as a result of the operation one image has become relatively more / less relevant than previously (for instance, when the patch masked a discriminative feature in that image).
- Another relevant SnapMix feature is **the ability to offer more effective augmentations as the training process progresses**. This is due to the fact that CAM works on top of the classifier, and therefore, becomes more accurate along with the classification improvement, making the resulting augmented samples more effective.

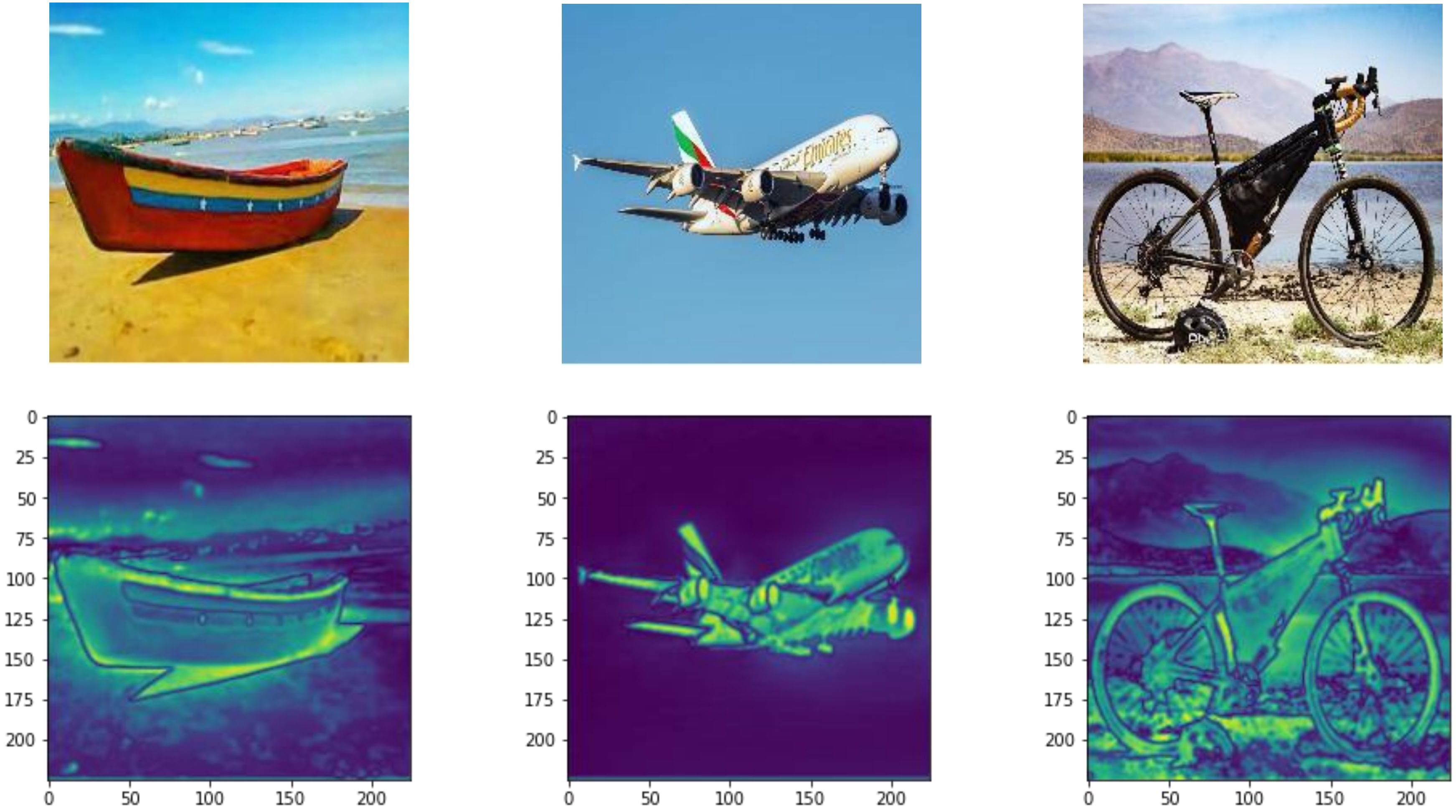
Saliency Mix

SALIENCY MIX - A F M Shahab Uddin, Mst. Sirazam Monira, Wheemyung Shin, TaeChoong Chung, Sung-Ho Bae (ICLR 2021). SaliencyMix: A Saliency Guided Data Augmentation Strategy for Better Regularization.



Source: own material

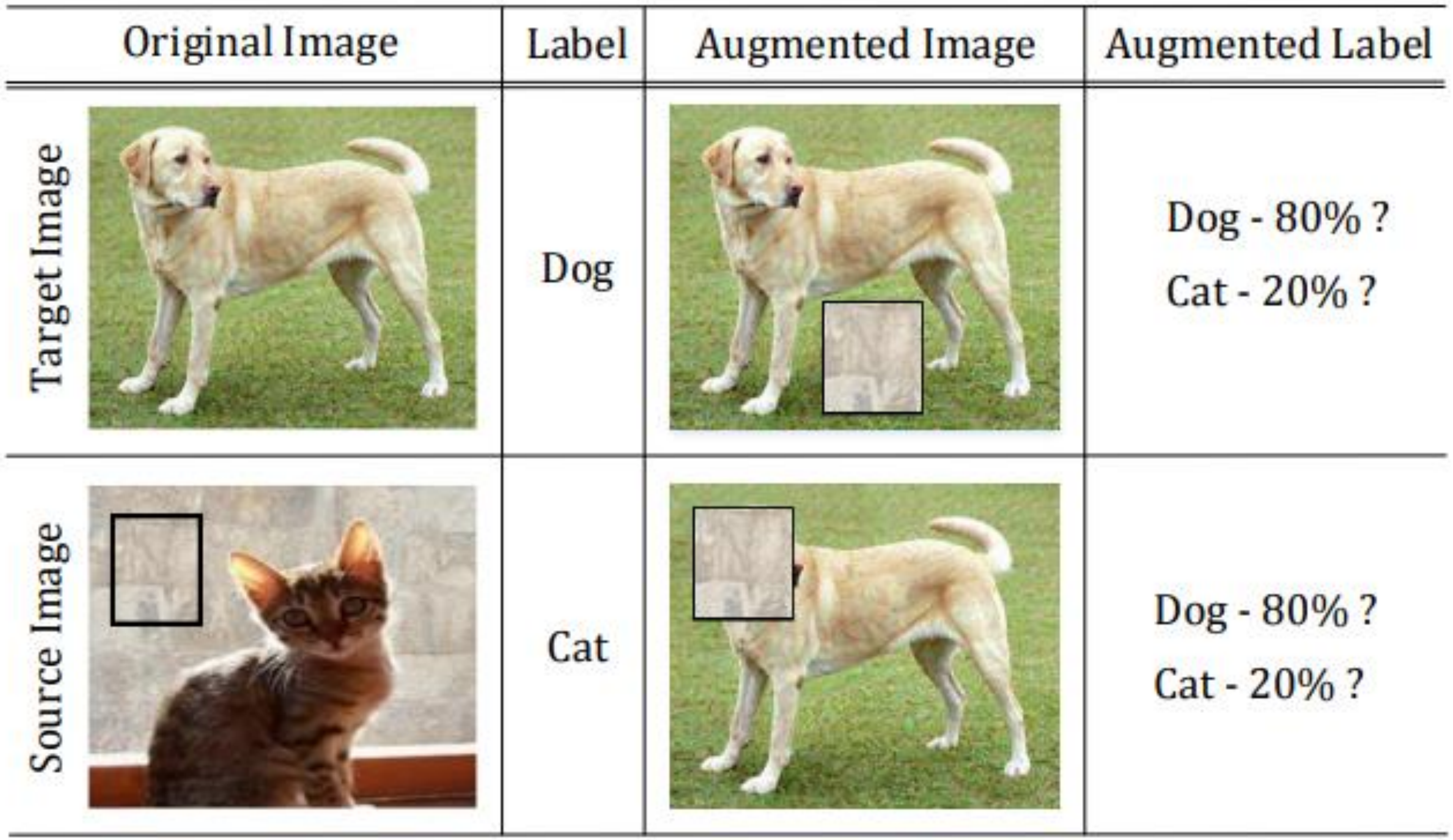
Saliency Calculation



https://colab.research.google.com/drive/1QpOAZX2E5Fq2N7MFET0bPlCUho2m3_X_?usp=sharing

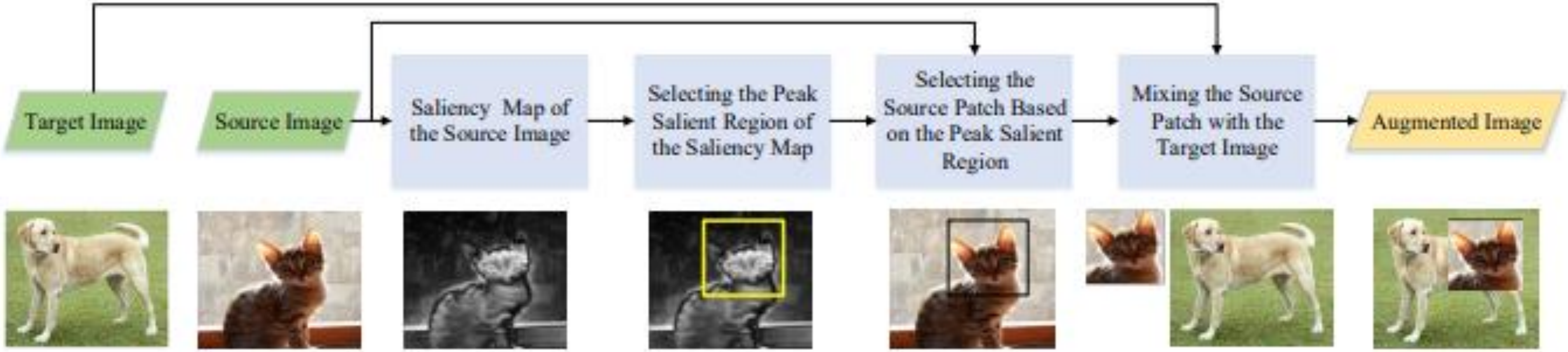
Saliency Mix

SALIENCY MIX - A F M Shahab Uddin, Mst. Sirazam Monira, Wheemyung Shin, TaeChoong Chung, Sung-Ho Bae (ICLR 2021). SaliencyMix: A Saliency Guided Data Augmentation Strategy for Better Regularization.



Saliency Mix

SALIENCY MIX - A F M Shahab Uddin, Mst. Sirazam Monira, Wheemyung Shin, TaeChoong Chung, Sung-Ho Bae (ICLR 2021). SaliencyMix: A Saliency Guided Data Augmentation Strategy for Better Regularization.



Saliency Mix

SALIENCY MIX - A F M Shahab Uddin, Mst. Sirazam Monira, Wheemyung Shin, TaeChoong Chung, Sung-Ho Bae (ICLR 2021). SaliencyMix: A Saliency Guided Data Augmentation Strategy for Better Regularization.

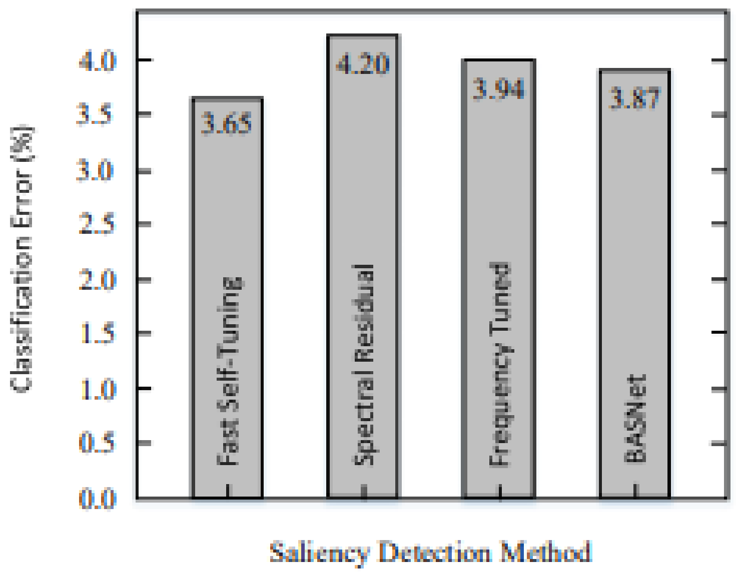


Figure 3. The effect of using different saliency detection methods on the proposed SaliencyMix data augmentation. Performances are reported from the average of five runs.

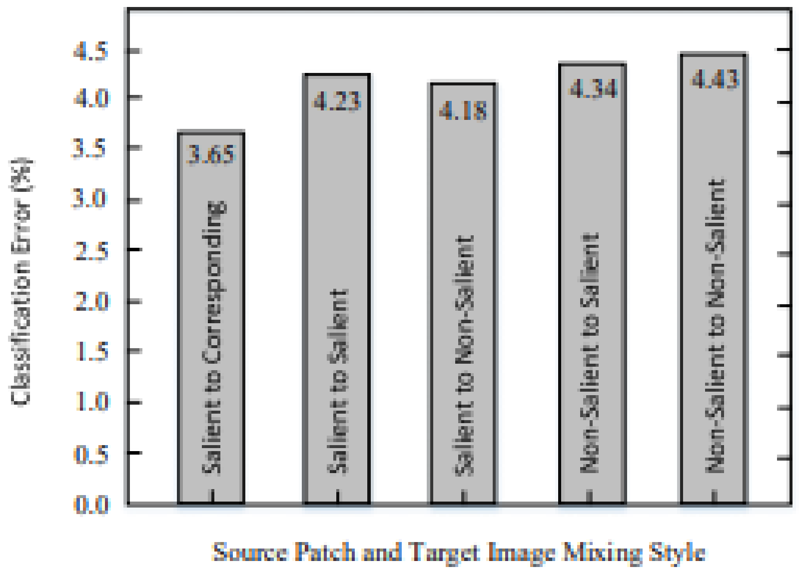


Figure 4. Different ways of mixing the source patch with the target image and their effects. Performances are reported from the average of five runs.

Saliency Mix

SALIENCY MIX - A F M Shahab Uddin, Mst. Sirazam Monira, Wheemyung Shin, TaeChoong Chung, Sung-Ho Bae (ICLR 2021). SaliencyMix: A Saliency Guided Data Augmentation Strategy for Better Regularization.

Table 1. Classification performance (average of five runs) of SOTA data augmentation methods on CIFAR-10 and CIFAR-100 datasets using popular standard architectures. An additional "+" sign after the dataset name indicates that the traditional data augmentation techniques have also been used during training.

METHOD	TOP-1 ERROR (%)			
	CIFAR-10	CIFAR-10+	CIFAR-100	CIFAR-100+
RESNET-18 (BASELINE)	10.63± 0.26	4.27± 0.21	36.68± 0.57	22.46± 0.31
RESNET-18 + CUTOUT	9.31±0.18	3.99±0.13	34.98±0.29	21.96±0.24
RESNET-18 + CUTMIX	9.44±0.34	3.78±0.12	34.42±0.27	19.42±0.23
RESNET-18 + SALIENCYMIX	**7.59±0.22**	**3.65±0.10**	**28.73±0.13**	**19.29±0.21**
RESNET-50 (BASELINE)	12.14±0.95	4.98±0.14	36.48±0.50	21.58±0.43
RESNET-50 + CUTOUT	8.84±0.77	3.86±0.25	32.97±0.74	21.38±0.69
RESNET-50 + CUTMIX	9.16±0.38	3.61±0.13	31.65±0.61	18.72±0.23
RESNET-50 + SALIENCYMIX	**6.81±0.30**	**3.46±0.08**	**24.89±0.39**	**18.57±0.29**
WIDERESNET-28-10 (BASELINE)	6.97±0.22	3.87±0.08	26.06±0.22	18.80±0.08
WIDERESNET-28-10 + CUTOUT	5.54±0.08	3.08±0.16	23.94±0.15	18.41±0.27
WIDERESNET-28-10 + CUTMIX	5.18±0.20	2.87±0.16	23.21±0.20	16.66±0.20
WIDERESNET-28-10 + SALIENCYMIX	**4.04±0.13**	**2.76±0.07**	**19.45±0.32**	**16.56±0.17**

Saliency Mix

SALIENCY MIX - A F M Shahab Uddin, Mst. Sirazam Monira, Wheemyung Shin, TaeChoong Chung, Sung-Ho Bae (ICLR 2021). SaliencyMix: A Saliency Guided Data Augmentation Strategy for Better Regularization.

Key points:

- Uses saliency information to **find the most representative pixels**
- The patch is either centered on the most salient pixel (if the entire patch fits within the image) or keeps this pixel within the patched area (otherwise).
- The authors considered four saliency calculation methods relaying either on statistical approaches or a learning based approach, and eventually decided to use the statistical saliency method as it offered slightly better classification accuracy
- **Invariant to image size**
- Various placements of the salient patch of the image were also tested, including placing the patch on salient / non-salient / corresponding part of the other image, with a conclusion that the choice of a **corresponding place is the best option in terms of regularization**.

Source: own material

Important patches detection on example of SaliencyMix vs. SnapMix

Vector	SaliencyMix	SnapMix
Algorithm used	Montabone & Soto	Class Activation Mapping
Type	Statistical method	Learning based model
Input image size	No limitations	Limited to the input size of the architecture used
Scope of important objects detection	No limitations	Limited to the classes the network was trained on

Source: own material

The end.
Thank you!

Discussion

Methods for identifying important signals in different modalities?