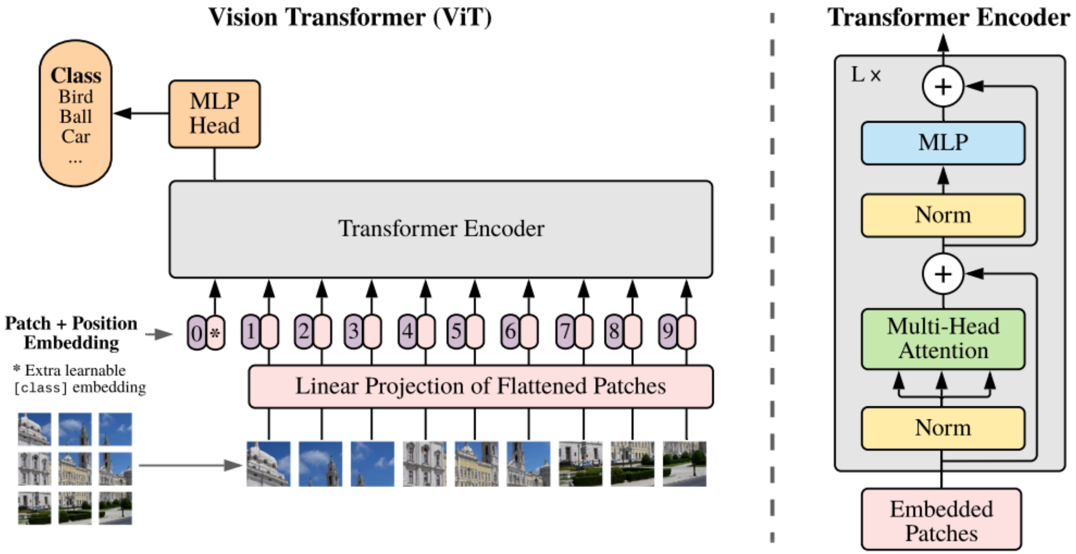
Transmix: Attend to mix for vision transformers

Chen, Jie-Neng, et al.
Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition. 2022.

Introduction
& Related Work & Methodology

Vision Transformer(ViT)

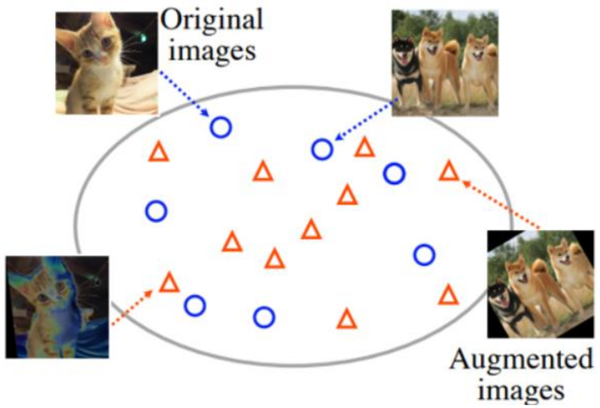
- Introduced into the field of computer vision and show great promise on tasks like image classification, object detection and image segmentation



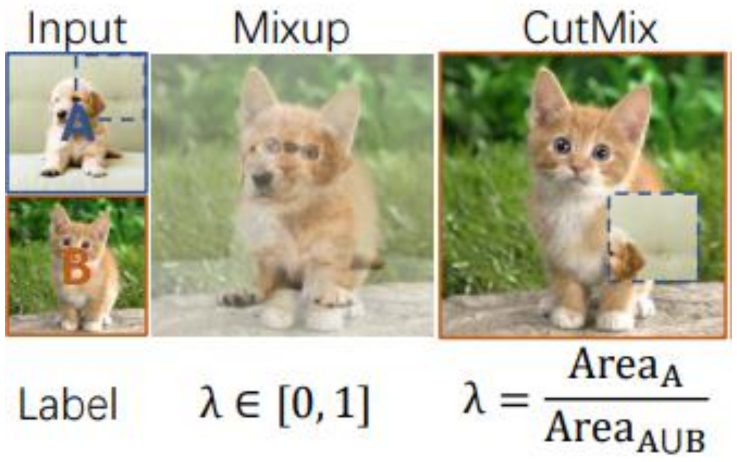
- **Limitations**
  - hard to optimize
  - can easily overfit if the training data is not sufficient

Solution

- **Apply augmentation and regularization techniques** to avoid overfitting to the training data

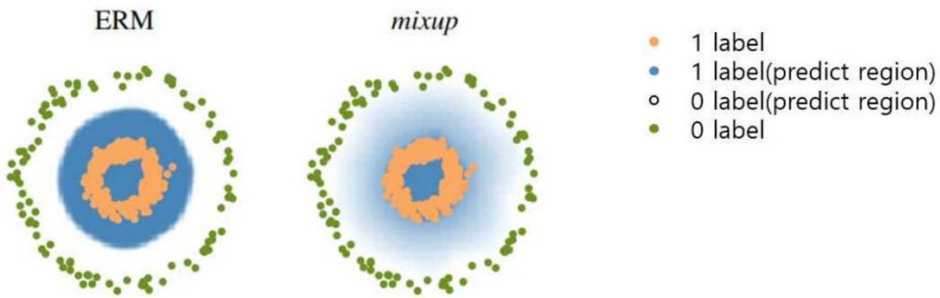


- **Mixup-based augmentation methods** on the input level

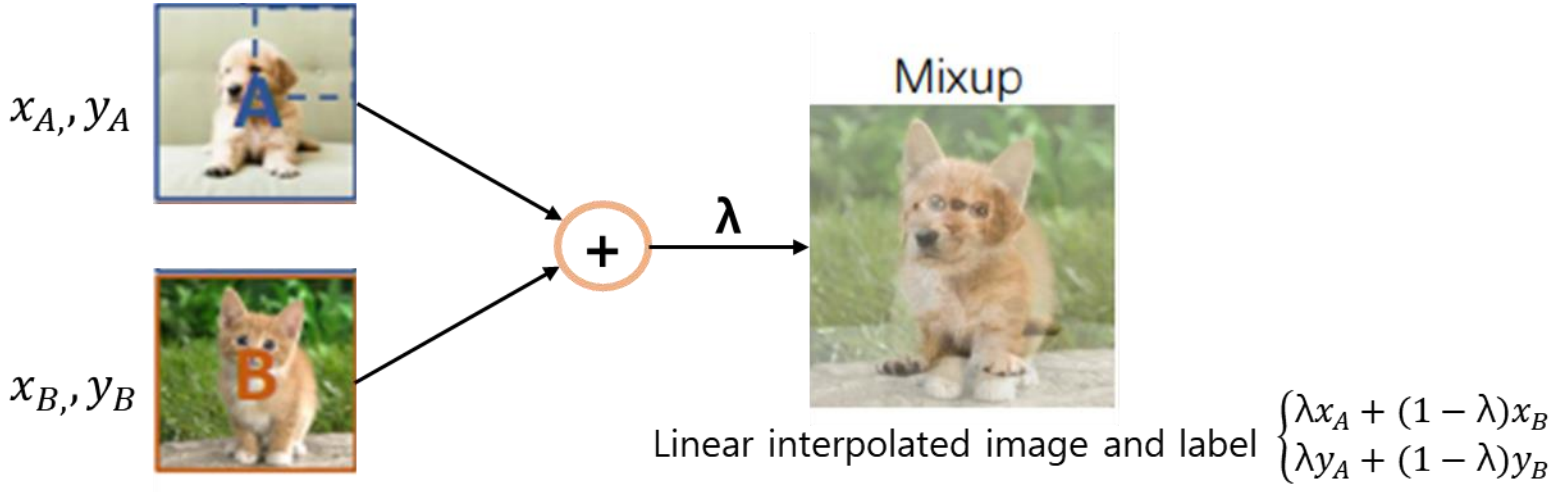


Zhang, Hongyi, et al.
"mixup: Beyond empirical risk minimization." arXiv preprint arXiv:1710.09412 (2017).

Mixup



- **Global image Mixture**
- **Pixel-wisely weighted combination of two global images**
  - A pair of inputs x_A, x_B and their corresponding labels y_A, y_B
  - λ: <u>Random mixing proportion sampled from Beta distribution</u>
  - **Pre-assume**: linear interpolations of feature vectors should lead to linear interpolations of the associated targets

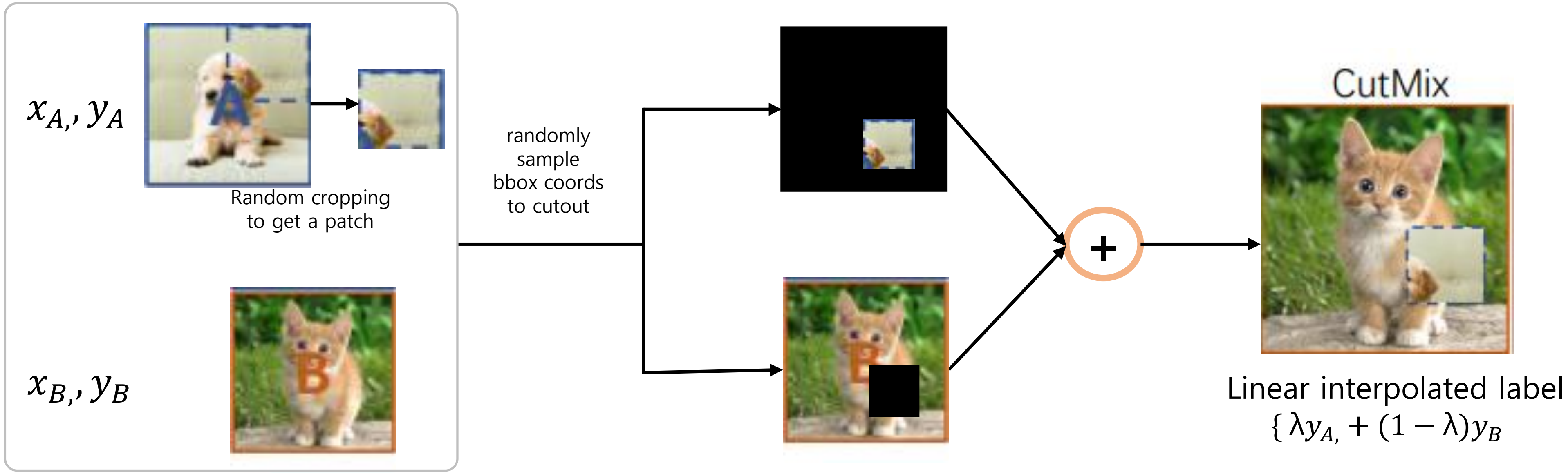


- **Limitations**
  - important features of the images can be diluted due to its simple linear combination approach

CutMix

- **Local image Mixture**

- $$\tilde{\mathbf{x}} = \mathbf{M} \odot \mathbf{x}_A + (\mathbf{1} - \mathbf{M}) \odot \mathbf{x}_B, \quad (1)$$
  $$\tilde{\mathbf{y}} = \lambda \mathbf{y}_A + (1 - \lambda) \mathbf{y}_B, \quad (2)$$
  - Binary mask M indicates the cutout and the fill-in regions from the two randomly drawn images
  - λ: equal to the cropped area ratio $\frac{r_w r_h}{WH}$



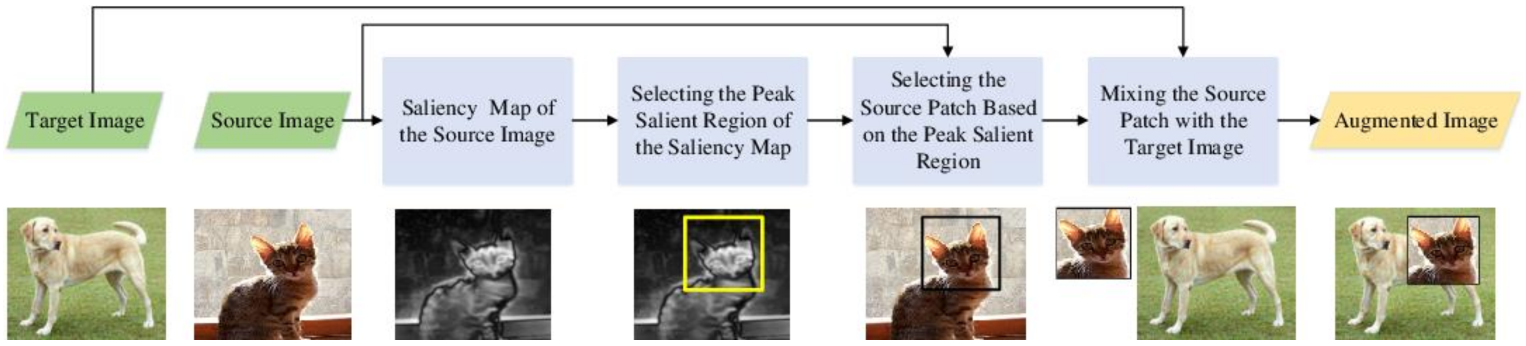
- **Limitations**
  - pixels in the background will not contribute to the label space as equally as those in the salient area

Uddin, A. F. M., et al. "Saliencymix: A saliency guided data augmentation strategy for better regularization." arXiv preprint arXiv:2006.01791 (2020).
Kim, Jang-Hyun, Wonho Choo, and Hyun Oh Song. "Puzzle mix: Exploiting saliency and local statistics for optimal mixup." International Conference on Machine Learning. PMLR, 2020.

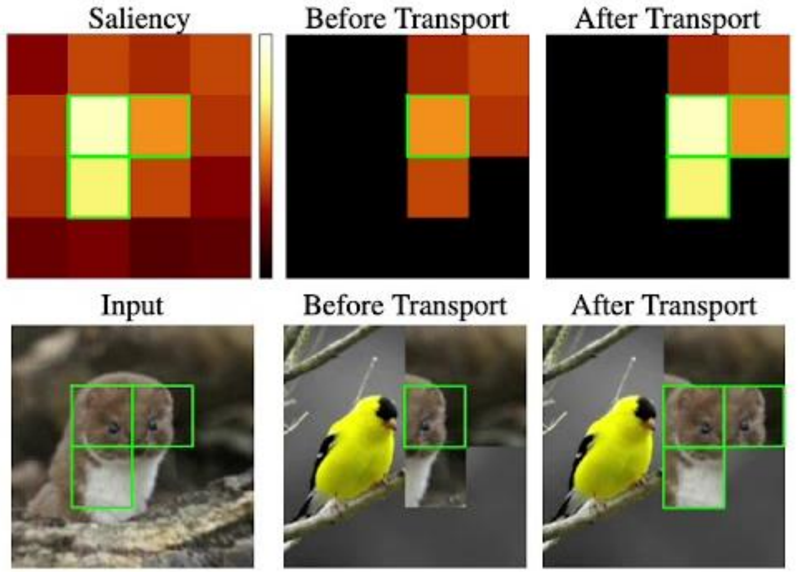
Saliency-based methods

- **Only mixing the most descriptive parts** on the input level

SaliencyMix



PuzzleMix



- **Limitations**
  - narrow the space of augmentation since they tend to less consider to put the background image into the mixture
  - cost more number of parameters and/or training throughput to extract the salient region of input.

Proposed method: TransMix

- **Leverage attention maps** that are naturally generated from ViTs.
  - mild the gap between the input and the label space through the learning of label assignment

x_A x_B

  - simply set λ (weight of y_A) as the **sum of weights of attention map lying in $\mathbf{A}$**

    ⇒ Labels are **re-weighted by the significance of each pixel**

- **Benefits**
  - can be merged into any Vit-based model training pipeline with no extra parameters and minimal computation overhead

TransMix

- Assign mixup labels with the guidance of attention map
  - The attention map is defined specifically as the multi-head class attention A
- In the Classification task,
  - Query q: class token
  - Key k: all input tokens
  - Class Attention A: the attention map from the class token to the input tokens
- Propose to use the class attention A to mix labels

Step 1) Multi-head Class Attention

1. Divide and embed an image $x \in R^{3 \times H \times W}$ to p patch tokens $x_{patches} \in R^{p \times d}$, and aggregate the global information by a class token $x_{cls} \in R^{1 \times d}$
   → ViTs operate on the patch embedding $[x_{cls}, x_{patches}] \in R^{(1+p) \times d}$

2. Parameterize the multi-head class attention with projection matrices $w_q, w_k \in R^{d \times d}$ and class attention for each head

$$\mathbf{q} = \mathbf{x}_{cls} \cdot \mathbf{w}_q, \tag{3}$$

$$\mathbf{k} = \mathbf{z} \cdot \mathbf{w}_k, \tag{4}$$

$$\mathbf{A}' = Softmax(\mathbf{q} \cdot \mathbf{k}^\top / \sqrt{d/g}), \tag{5}$$

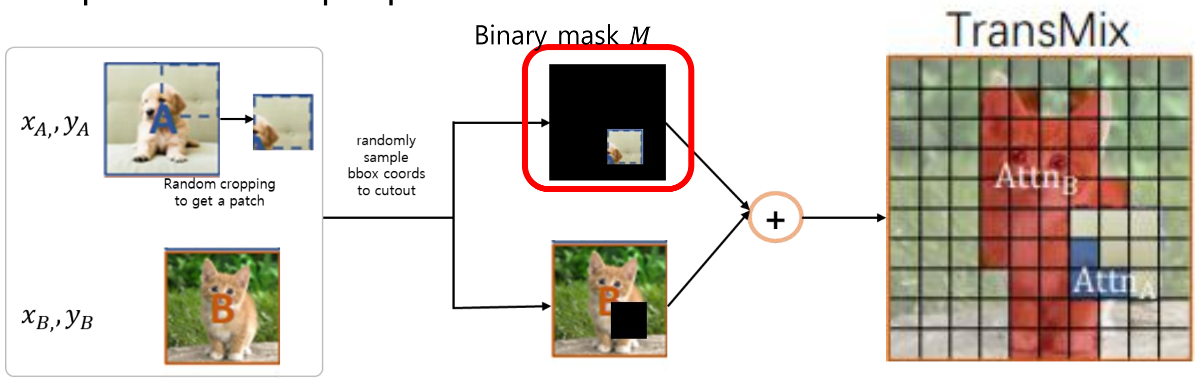
$$\mathbf{A} = \{\mathbf{A}'_{0,i}, \mid i \in [1, p]\}, \tag{6}$$

   – $A \in [0,1]^p$ is the attention map from the class token to the image patch tokens, summarizing which patches are the most useful to the final classifier

   – Simply average across all attention heads to obtain $A \in [0,1]^p$

Step 2) Mixing labels with the attention map

1. Follow the process of input mixture proposed in CutMix



$\{\lambda y_{A,} + (1-\lambda) y_B$

2. Calculate λ

$$\lambda = \mathbf{A} \cdot \downarrow (\mathbf{M}). \qquad (7)$$

– ↓ (·) denotes the nearest-neighbor interpolation downsampling that can transform the original M from HW into p pixels .

(Note that we omit the dimension unsqueezing in Eqn. (7) for simplicity)

– Network can learn to re-assign the weight of labels for each data point dynamically based on their responses in the attention map

Pseudo-code

Algorithm 1 Pseudocode of TransMix in a PyTorch-like style.

```
# H, W: the height and width of the input image
# p: number of patches
# M: 0-initialized mask with shape (H,W)
# downsample: downsample from length (H*W) to (p)
# (bx1, bx2, by1, by2): bounding box coordinate

for (x, y) in loader: # load a minibatch with N pairs
    # CutMix image in a minibatch
    M[bx1:bx2, by1:by2] = 1
    x[:,:,M==1] = x.flip(0)[:,:,M==1]
    M = downsample(M.view(-1))

    # attention matrix A: (N, p)
    logits, A = model(x)

    # Mix labels with the attention map
    lam = matmul(A, M)
    y = (1-lam) * y + lam * y.flip(0)

    CrossEntropyLoss(logits, y).backward()
```

Experiment

- Paper