# 1. 核心算法：

## （1） 文中的核心算法为CutBlur算法。

**a. 算法理解：**

首先我们给出两个图像补丁$LR: x_{LR} \varepsilon R^{W \times H \times C}, LR: x_{HR} \varepsilon R^{sW \times sH \times C}$，其中表示SR中的比例因子。如图1所示，因为CutBlur需要匹配$x_{LR}$和$x_{HR}$的分辨率，我们首先使用双三次内核来查询$x_{LR}$ s次，即$x_{LR}^{s}$。CutBlur的目标是生成一对新的训练样本($\hat{x}_{HR} \rightarrow LR$，$\hat{x}_{LR} \rightarrow HR$)，通过剪切和粘贴$x_{HR}$的随机区域到相应的$x_{LR}^{s}$中，反之亦然:

$$\begin{aligned} \hat{x}_{HR \rightarrow LR} &= \mathbf{M} \odot x_{HR} + (\mathbf{1} - \mathbf{M}) \odot x_{LR}^{s} \\ \hat{x}_{LR \rightarrow HR} &= \mathbf{M} \odot x_{LR}^{s} + (\mathbf{1} - \mathbf{M}) \odot x_{HR} \end{aligned} \tag{1}$$

where $\mathbf{M} \in \{0,1\}^{sW \times sH}$ denotes a binary mask indicating where to replace, $\mathbf{1}$ is a binary mask filled with ones, and $\odot$ is element-wise multiplication. For sampling the mask and its coordinates, we follow the original CutMix [36].

$\mathbf{M} \in \{0,1\}^{sW \times sH}$ 表示二值化mask。

对此，我的理解是：通过将两张分辨率一高一低的图像进行处理，模拟训练模型，通过不断寻找图像缺失位置，将低分辨率图像粘贴到高分辨率图像上，反过来也是这样。

**b. 突出创新点：**

CutBlur在具有相同内容的LR和HR图像补丁之间进行剪切和粘贴。通过将LR缝补到HR上，而由于图像内容不匹配，cutblur能够最小化边界效应。同时，cutblur利用的是整个图像的信息，同时随机的HR比和位置的样本不同，它具有正则化效果。

同时CutBlur也能注意到传统DA方法运用会破坏图像的空间信息，因此其注重对图像空间信息的保护。

# 2. 功能模块

## （1）像素空间中的DA方法：

包括Mixup、Cutup、AutoAugment（最佳增强策略）等方法。这些方式均可以用于高级视觉任务。

很多研究在高级视觉视觉任务中增强图像，如下：

$Mixup$ :混合两张图片，生成副本

$Cutout$ :切掉图像中的某个部分，但这样会导致其无法充分利用数据集，因此我们可以在其空出的区域添加一个额外的图像。

$AutoAugment$ :被认为是最佳增强策略，用于学习给定任务和数据集。其能够创建一个数据增强策略的搜索空间，利用搜索算法选取适合特定数据集的数据增强策略。此外，从一个数据集中学到的策略能够很好地迁移到其它相似的数据集上。（https://blog.csdn.net/pwtd_huran/article/details/80868435）

## （2） 特征空间的DA方法：

此类方法能够操纵CNN特征，包括：特征混合、抖动、下降。

以及提出能够操纵CNN特征的DA方法，分为三类：特征混合、抖动、下降。

特征混合：此方法将输入图像和潜在特征混合（例如(Java：拉普拉斯锐化、Sobel边缘检测、均值滤波、伽马变换）；

抖动：对特征执行随机仿射变换（其实就是利用双线性插值的方式对图像进行处理）

下降：对多余的特征有选择性的删除，提高模型泛化能力。

## （3） 超分辨率下的DA方法

简单几何操作：翻转，旋转。多用于SR模型中。这里提出Mixup可以缓解SR模型的过拟合问题。

## （4） 综合分析：

将以上方法进行综合利用，结合起来对图像进行处理，能够提高cpu的利用率、提高图像恢复的准确率。我们可以通过作者的实验结果进行分析：我们通过观察下图，通过不同的方式随机挖去图片的一部分像素，通过输入恢复图象时，性能的降低也会有所不同。例如去除25%的矩形形状内容，其在Cutout下的性能降低%1，然而恢复后，像素 却增加0.01和0.06。

Table 1. PSNR (dB) comparison of different data augmentation methods in super-resolution. We report the baseline model (EDSR [18]) performance that is trained on DIV2K (×4) [2] and RealSR (×4) [5]. The models are trained from scratch. $\delta$ denotes the performance gap between with and without augmentation.

| Method | DIV2K ($\delta$) | RealSR ($\delta$) |
|---|---|---|
| EDSR | 29.21 (+0.00) | 28.89 (+0.00) |
| Cutout [8] (0.1%) | 29.22 (+0.01) | 28.95 (+0.06) |
| CutMix [36] | 29.22 (+0.01) | 28.89 (+0.00) |
| Mixup [37] | 29.26 (+0.05) | 28.98 (+0.09) |
| CutMixup | 29.27 (+0.06) | 29.03 (+0.14) |
| RGB perm. | 29.30 (+0.09) | 29.02 (+0.13) |
| Blend | 29.23 (+0.02) | 29.03 (+0.14) |
| CutBlur | **29.26 (+0.05)** | **29.12 (+0.23)** |
| All DA's (random) | **29.30 (+0.09)** | **29.16 (+0.27)** |

## 3. 输入输出：

**CutBlur vs 训练使用HR输入**：CutBlur性能更好，因为M=0/1包含了后者的情况。而且CutBlur教会模型更好学习where去超分。

**混合增广MoA**：为获得最佳性能，我们混合多种DA。对每轮训练，模型首先用概率p决定是否用DA。然后随机选一个DA方法。

**对于输入的要求**：我们所需要的输入为一张图片，图片应当是从某个图像中所取出的一小部分（LR低质量图片），或者是（HR超分辨图片）挖去一块后剩下的部分。

**对于输出的要求**：我们使用类似于感温器的模型来输出图像，从中可以清晰的看出图像运行后的锐化效果与边界效应。

在输入输出图片时，应当保证挖去部分与所需填充的部分空间信息一致，以降低边界效应的影响 。