

# 创建自定义图像Inpainter

Inpaint 是一种用于填充图像中缺失值的通用函数。如果图像区域是特定的，则可以使用自定义图像 inpainter 来获得更好的结果。下面的例子说明如何使用 LearnDistribution 和 SynthesizeMissingValues 创建这样的自定义 inpainter。

获取图像数据集。


```
In[*]:= data = ResourceData["MNIST", "TrainingData"] [[All, 1]];
           |资源数据 |全部

In[*]:= RandomSample[data, 10]
           |伪随机采样

Out[*]:= {7, 1, 4, 2, 4, 1, 5, 3, 0, 4}
```

在图像的像素值上训练分布。

```
In[*]:= ld = LearnDistribution[ImageData /@ data, Method -> "KernelDensityEstimation"]
           |学习分布 |图像数据 |方法

Out[*]:= LearnedDistribution[
   Input type: NumericalTensor (size: 28 x 28)
  Method: KernelDensityEstimation
  数据存储在 LocalObject[...]
```


填充新图像的缺失值（由灰色像素定义）。

```
In[*]:= Image /@ SynthesizeMissingValues[ld, ImageData /@
           |图像 |合成丢失的值 |图像数据

           {7, 1, 4, 2, 4, 1, 5, 3, 0, 4}, MissingValuePattern -> 0.5]
           |丢失值得模式

Out[*]:= {7, 4, 4, 0, 0, 7, 6, 3, 7}
```

与一般 Inpaint 函数的结果进行比较。

```
In[*]:= Inpaint[#, ] & /@ {7, 1, 4, 2, 4, 1, 5, 3, 0, 4}
           |润饰

Out[*]:= {7, 1, 4, 0, 0, 7, 6, 3, 7}
```