

医学超声图像去噪修复与增强

IE308 Final Project, December 2019.

*Name: Zixuan Li (李紫轩) 319830067@qq.com

*Name: Mingke Wang (王铭珂) wangmingke@sjtu.edu.cn

目录

1 项目内容	2
2 基于传统图像处理方法的实现	2
2.1 图像内容分析与特征提取	2
2.1.1 色彩空间滤波和硬阈值滤波	2
2.1.2 图像形态学处理	3
2.2 基于纹理合成 Inpainting 算法 [1] 的图像复原	3
2.2.1 优先级计算	4
2.2.2 暴力搜索	5
2.3 算法设计思路	5
2.4 基于经典算法的实验结果分析	7
3 基于 DnCNN 神经网络 [3] 的去噪处理算法实现	7
3.1 DnCNN 网络模型	7
3.2 残差学习 (residual learning)	8
3.3 批量标准化 (batch normalization)	8
3.4 基于 DnCNN 神经网络的实验结果分析	9
4 对传统图像处理方法和基于神经网络的学习算法的分析与比较	10

1 项目内容

医学超声成像技术因为其非侵入性、无辐射、实时性以及高性价比等特点，已经被广泛应用于医疗辅助诊断当中。但医生在保存超声图像过程中可能会在图上留下人工标记（artifacts），如图 1 和图 2 所示。

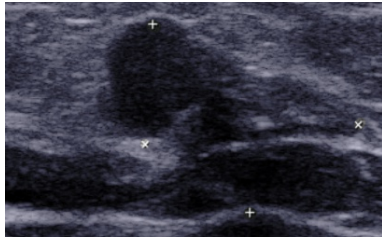


图 1: 带有 artifacts 的图像

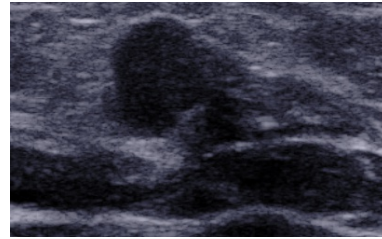


图 2: 无 artifacts 的图像

另外，灰阶超声图像容易受患者呼吸等外界干扰导致图像出现噪声和伪影，图像质量普遍较差。在本次项目中，我们分别应用传统图像处理方法和基于神经网络的机器学习算法对医学超声图像的去噪修复提出了效果显著的解决办法。同时，在本文中我们还会对传统图像算法与机器学习算法的效果与性能进行仔细的分析比较。

2 基于传统图像处理方法的实现

对于 artifacts 的去除，整体算法可以分为两个部分。首先是对于 artifacts 部分的识别、提取与删除，其次是对被删除 artifacts 部分的图像复原与填充。

2.1 图像内容分析与特征提取

2.1.1 色彩空间滤波和硬阈值滤波

通过对大量数据集中带噪声图像的分析，我们可以发现图像中带有 artifacts 部分与其余部分图像色阶相差明显，有些图像的 artifacts 还带着明显的黄色或蓝色，所以我们尝试在不同的色彩空间对图像进行提取，观察在不同色彩通道中图像的表现。

首先我们尝试了在 RGB 色彩通道下对图像进行拆分和输出，效果不是非常明显。最终我们发现对于具有某种目标色的像素，可以直接通过对 H（色调，Hue）、S（饱和度，Saturation）、V（明度，Value）三分量设定一个范围来把它们过滤出来，无需进行较复杂的色彩距离计算，这样可以在色彩分割时节省大量的时间。这种过滤对我们对象数据集中的超声图像上的标记特别有效，因为 HSV 色域对于黑色和白色的过滤效果不明显。这是由于对于纯色的黑色和白色，它们的色调和饱和度没有意义，所以和其它颜色相比缺少了两个过滤指标。我

们在 H 空间进行硬阈值滤波，选取合适的阈值，就可以有效地对标记点进行提取和识别。

特征提取成功之后，我们将提取出的区域在原图上作出 **mask**，也就是在相应的区域置零。

2.1.2 图像形态学处理

在进行了色彩空间滤波和硬阈值滤波之后，我们发现 **mask** 图像上仍然有一些小的杂点，我们当然想要尽可能保留原图的信息，并且我们希望把 **artifacts** 尽可能的去除。由此，我们对经过上述滤波之后的图像进行了图像形态学的处理，包括开和闭操作。开和闭操作都包含了两个相同的步骤：腐蚀和膨胀，但执行顺序不同。开操作是先腐蚀后膨胀，闭操作是先膨胀后腐蚀。膨胀操作，具体方法如下：

1. 用 **kernel B** 扫描图像 **A** 的每一个元素；
2. 用 **kernel** 与其覆盖的二值图像做 **and** 操作；
3. 如果都为 0，则结果图像的该像素为 0，否则为 1。

同理，要去除被误判为 **artifacts** 的更小的杂点，我们采用腐蚀的具体方法如下：

1. 用 **kernel C** 扫描图像 **A** 的每一个元素；
2. 用 **kernel** 与其覆盖的二值图像做 **and** 操作；
3. 如果都为 1，则结果图像的该像素为 1，否则为 0。

我们定义了一个 7x7 的 **kernel B**（与图像上 **artifacts** 大小大致相同）对图像进行了闭操作，同理，要去除被误判为 **artifacts** 的更小的杂点，我们采用闭操作的方法，首先定义一个 3x3 的 **kernel C**（图中小于 3x3 的点均会被腐蚀）对图像进行闭操作。经过上述处理，可以使得 **mask** 的效果更加精确。

2.2 基于纹理合成 Inpainting 算法 [1] 的图像复原

接下来就是对已经做过 **mask** 操作的图片进行空白区域的填充。在 2004 年 Criminisi 等人在 [1] 中将纹理合成算法进一步拓展，使其可以用于更广泛的自然图像，而不仅仅是纹理图像。该算法可以去除大块的前景像素，去除杂乱的信息并填充图像中缺失的像素。在本节我们将针对此算法进行介绍。

图 3 形象地说明了该算法的执行过程。(a) 代表原图，其中 Ω 区域代表目标区域 (target region)，即将要被填充的区域， $\delta\Omega$ 代表目标区域的轮廓。 Φ 代表源区域 (source region)。(b) 我们希望合成以点 $\mathbf{p} \in \delta\Omega$ 为中心的块 (patch)

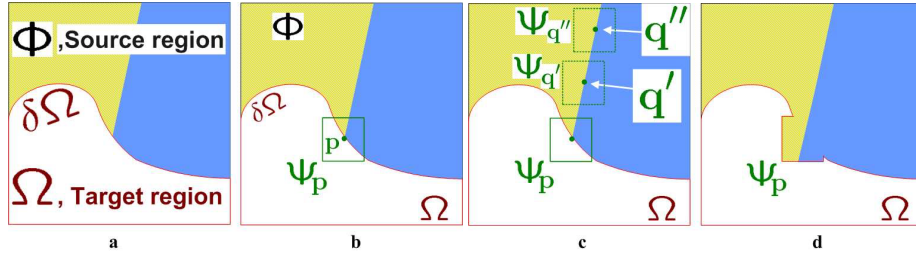


图 3: 纹理合成算法示意图 [1]

Ψ_p 。(c) Ψ_p 的最佳匹配块坐落于源区域 Φ 两种材质之间的交界处上的的块，比如 $\Psi_{q'}$ 和 $\Psi_{q''}$ 。(d) 候选匹配块集合中的最佳匹配块已经被填充到块 Ψ_p 上，部分填补了 Ω 区域的空缺。需要注意的是，纹理和结构都被传递到了之前的目标区域 Ω 中，并且 Ω 区域和它的轮廓 $\delta\Omega$ 现在已经缩减到了不同的形状。

该算法主要由优先级计算，搜索及复制三个部分组成。

1. 优先级决定了移除顺序，保证图像中的线性结构传播，目标边界连通；
2. 搜索即根据纹理相似距离 (平方差距离和, SSD, Sum of Squared Difference) 在已知区域中找到最佳匹配块；
3. 复制即将最佳匹配块复制到对应的目标区域位置。

2.2.1 优先级计算

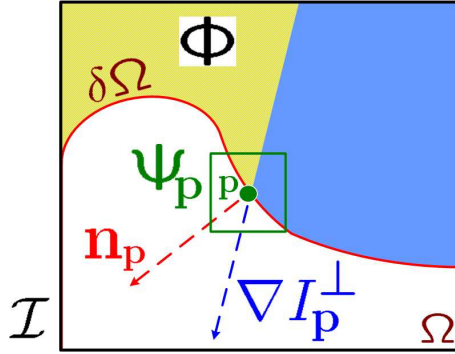


图 4: 计算方法示意图 [1]

如图 4 所示，边界上有很多像素点，以这些点为中心可以得到很多 patch (比如 9×9 的 patch)，对于这些 patch，都需要计算一下他们的优先级是多少，以便从中选出一个优先级最大的块作为首要修复的对象。对于一个 patch，块的优先级计算公式：

$$P(\mathbf{p}) = C(\mathbf{p}) * D(\mathbf{p}) \quad (1)$$

其中， \mathbf{p} 代表这个待修复块的中心点， $C(p)$ 表示块的置信度， $D(p)$ 为 data term，即数据项。置信度的计算公式为：

$$C(\mathbf{p}) = \frac{\sum_{\mathbf{q} \in \Psi_{\mathbf{p}} \cap (\tau - \Omega)} C(\mathbf{q})}{|\Psi_{\mathbf{p}}|} \quad (2)$$

其中 I 为整张图像， Ω 为 I 的缺失区域， $\Psi_{\mathbf{p}}$ 为待填充的块， $|\Psi_{\mathbf{p}}|$ 为块的面积。 $c(q)$ 为块中的像素点的置信度，在初始化的时候，已存在的像素点的置信度为 1，缺失的像素点的置信度为 0，在修复过程中，修复出来的像素点的置信度被更新为块的置信度，由此可知，随着修复过程的推进，修复出来的像素点的置信度会越来越小。

数据项（data term）的计算公式为：

$$D(\mathbf{p}) = \frac{|\nabla I_{\mathbf{p}}^{\perp} \cdot \mathbf{n}_{\mathbf{p}}|}{\alpha} \quad (3)$$

这里是等照度向量（isophote） $\nabla I_{\mathbf{p}}^{\perp}$ 与法向量 $\mathbf{n}_{\mathbf{p}}$ 的点乘再求模长， α 为归一化因子。在计算等照度向量时需要先计算点的 image gradient，等照度线向量为 gradient 逆时针旋转 90° ，gradient 代表了像素值变化最快的方向，而等照度线向量与 gradient 垂直，代表了变化最慢的方向。

如下给出 image gradient 的计算公式：

$$\nabla f = \begin{bmatrix} g_x \\ g_y \end{bmatrix} = \begin{bmatrix} \frac{\partial f}{\partial x} \\ \frac{\partial f}{\partial y} \end{bmatrix} \quad (4)$$

2.2.2 暴力搜索

计算出每一个块（patch）的优先级之后，从中选出一个优先级最高的 patch 作为待修复对象，如。再使用暴力法，从源区域中选出一个与之最相近的块，将中对应的像素拷到中（这里只需要修复中缺失的像素点）。使用暴力法的时候，衡量两个块之间的距离时，使用 SSD（差的平方和）作为距离，由于本身就缺失了部分像素值，所以计算的就是非缺失像素和其他 patch 对应位置像素值的 SSD。

需要注意的是，在修复过程的迭代中，缺失区域逐渐变小，但是提供像素来源的目标区域固定不变。因为目标区域中，修复出来的像素值可信度比较低，不予以采用。

2.3 算法设计思路

具体的算法实现思路如下所示：

Algorithm 1: 图像内容分析与特征提取

Data: 带噪声医学超声图像

Result: 带 mask（噪声上）图像

初始化：读入带 artifacts 图像；

将图像从 RGB 色彩空间转到 HSV 色彩空间；

提取出 H 通道的图片层；

设定阈值；

if 大于阈值 **then**

 数组值置为 255；

else

 数组值置为 0；

end

对二值化的图像用 7x7 的 kernel 进行闭操作；

对二值化的图像用 3x3 的 kernel 进行开操作；

再用 3x3 的 kernel 对图像进行膨胀处理；

得到二值化图像和原图相叠加，输出结果；

Algorithm 2: 根据图像目标区域边缘中块的优先级对图像进行填充

Data: 带 mask（噪声上）图像

Result: 复原出的图像

初始化，读入图像；

提取目标区域 Ω 的边缘 $\delta\Omega$ ；

while 边缘不为空 **do**

 计算边缘 $\delta\Omega$ 上所有块的优先级 $P(p)$ ；

 找到优先级最高的块 Ψ_p ；

 从源区域 Φ 中找到块 Ψ_q ，使得 $d(\Psi_p, \Psi_q)$ 最小；

 将 Ψ_q 中的数据复制到 Ψ_p 中的目标填充点 p ；

 更新 $C(p)$ ，更新边缘 $\delta\Omega$ ；

end

2.4 基于经典算法的实验结果分析

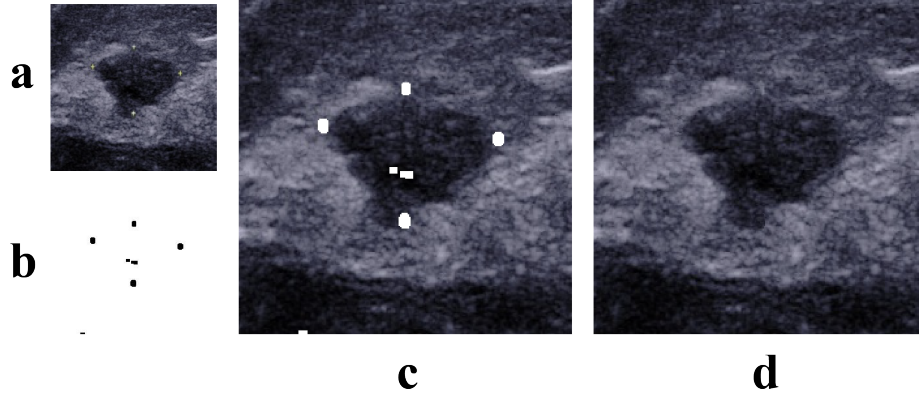


图 5: 实验结果 1

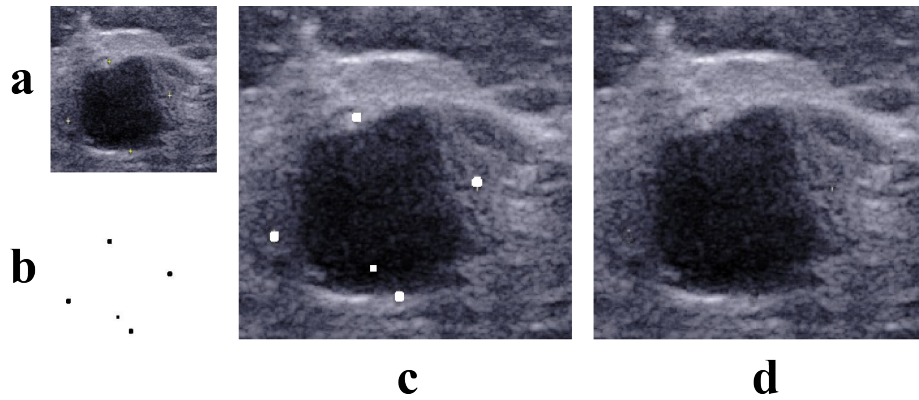


图 6: 实验结果 2

实验结果罗列如图所示，其中 (a) 是带有 artifacts 的超声图像，(b) 是经过色彩空间滤波、硬阈值滤波和图像形态学处理后的二值化 mask 图像，(c) 是在 (a) 的基础上挖去定位到的 mask 部分，最后 (d) 是复原后的图像。

通过结果可以看出，图像上 artifacts 定位的非常准确，完全去除了人工标记；基于 Inpainting 图像复原算法的填充操作的效果可以说没有任何不自然的痕迹。

3 基于 DnCNN 神经网络 [3] 的去噪处理算法实现

3.1 DnCNN 网络模型

网络结构

第一部分: $Conv(3 * 3 * c * 64) + ReLu$ (c 代表图片通道数)

第二部分: $Conv(3 * 3 * 64 * 64) + BN(batchnormalization) + ReLu$

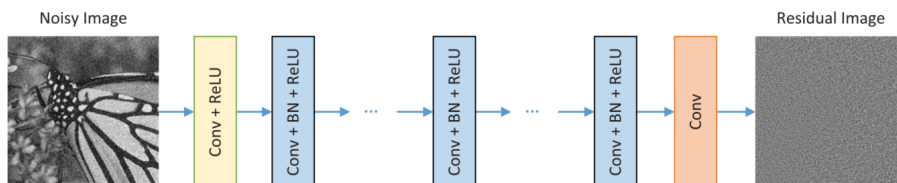


图 7: DnCNN 网络结构 [3]

第三部分: $Conv(3 * 3 * 64)$

每一层都 **zero padding**, 使得每一层的输入、输出尺寸保持一致。以此防止产生人工边界 (**boundary artifacts**)。第二部分每一层在卷积与 **ReLU** 之间都加了批量标准化 (**batch normalization, BN**)。

3.2 残差学习 (residual learning)

DnCNN 结合了 ResNet 的 **residual learning**[2]。不同的是 DnCNN 并非每隔两层就加一个 **shortcut connection**, 而是将网络的输出直接改成 **residual image** (残差图片), 设纯净图片为 x , 带噪音图片为 y , 假设 $y=x+v$, 则 v 是残差图片。即 DnCNN 的优化目标不是真实图片与网络输出之间的 **MSE**(均方误差), 而是真实残差图片与网络输出之间的 **MSE**。

根据 ResNet 中的理论, 当残差为 0 时, 堆积层之间等价于恒等映射, 而恒等映射是非常容易训练优化的。作者注意到在图像复原领域 (尤其是在噪音程度较小的情况下), 噪音图片与纯净图片的残差非常小, 所以理论上残差学习非常适合运用到图像复原上。

通俗来讲, 这样的网络设计就是在隐层中将真实的图片 x 从原噪音图 y 中消去。作者注意到: 在超分领域, 低分辨率图片就是高分辨率图片的双三次上采样操作形成的, 故超分领域的残差图片和去高斯噪声领域的残差图片是等价的, 同理还有 **JPEG** 解锁领域的残差图片。也么一来, 用一个模型应对三种问题便有了可能性, 最终实验证明确实有效。

3.3 批量标准化 (batch normalization)

SGD(随机梯度下降法) 广泛应用于 CNN 的训练方法中, 但是训练的性能却很大程度受内部协变量移位这一问题所影响。**BN** 就是在每一层的非线性处理之前加入标准化、缩放、移位操作来减轻内部协变量的移位。可以给训练带来更快的速度, 更好的表现, 使网络对初始化变量的影响没有那么大。

内部协变量移位 (internal covariate shift): 深层神经网络在做非线性变换前的激活输入值, 随着网络深度加深或者在训练过程中, 其分布逐渐发生偏移或者变动, 之所以训练收敛慢, 一般是整体分布逐渐往非线性函数的取值区间的上下限两端靠近 (对于 **Sigmoid** 函数来说, 意味着激活输入值 $WU+B$ 是大的

负值或正值)，所以这导致反向传播时低层神经网络的梯度消失，这是训练深层神经网络收敛越来越慢的本质原因。

批量标准化 (batch normalization)：就是通过一定的规范化手段，把每层神经网络任意神经元这个输入值的分布强行拉回到均值为 0 方差为 1 的标准正态分布，即把越来越偏的分布强制拉回比较标准的分布，这样使得激活输入值落在非线性函数对输入比较敏感的区域，所以输入的小变化才就会导致损失函数有较大的变化，意思就是让梯度变大，避免梯度消失问题产生，而且梯度变大意味着学习收敛速度快，能大大加快训练速度。

3.4 基于 DnCNN 神经网络的实验结果分析

对于神经网络，我们所用的数据集包括：来自不同病人的 100 对（200 张，包括带噪声图和原图）灰阶超声图像作为算法训练集，五十张待复原图像作为算法评估集。

以下给出在算法测试集中的表现，我们选取 epoch384 的模型作为验证模型。

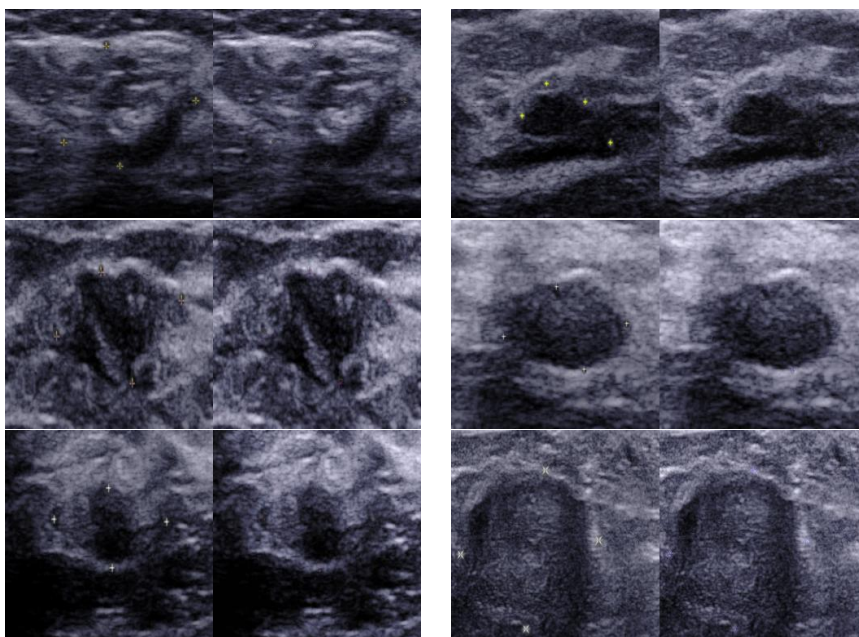


图 8: 测试集训练结果

由图 8（左侧是带有 artifacts 的噪声图，右侧是复原出的图片）我们可以发现，网络模型可以完全干净地去除超声图像中的 artifacts，并对带噪声图像进行了调整白平衡和去噪的处理，经过对比可以发现原图偏黄并带有浑浊的色彩，而训练出的结果图色彩均衡、更加清晰。

4 对传统图像处理方法和基于神经网络的学习算法的分析与比较

对图像处理的传统算法而言，如果想对图像进行分类，首先需要执行一个特征提取的步骤。特征是图像中比较小的“有趣的”、具有描述性的或包含信息的块。就特征提取和图像分类而言，使用这些技术的思路是：从统一类别对象的图像中（椅子、马等）提取尽可能多的特征，并将这些特征视为对象的一个“定义”（众所周知的词袋模型），然后在其他图像中搜索这些“定义”，如果词袋模型中有相当一部分的特征都可以在这幅图像中找到，那么这幅图像被分类为包含该特定对象的类别（椅子、马等）。只有使用不同的特征才可以更好地描述不同类别的对象。如果你在分类时使用很多特征，就必须对大量的参数进行微调。

在我们的实验中，由于测试集中全都是类似十字点的 **artifacts** 标记，所以只需要采取一种特征提取的方法。但是如果这些 **artifacts** 更加的多样化，那么针对不同的 **artifacts**（点、线、圈等等）我们就要选取不同的方法，不同的参数来对其进行提取。

深度学习为我们展示了端到端学习（**end-to-end learning**）这一概念，简而言之，针对每个特定类别的对象，机器会自动学习需要查找什么特征。它为每个对象提供了最具描述性和显著性的特征。换句话说，神经网络可以探索图像类别中的底层模式。因此，通过端到端的学习，你不再需要自己动手来决定使用哪种传统计算机视觉技术来描述这些特征，机器将会替你做这些工作。

但深度学习的训练模型也有其弊端，首先它需要大量的数据，同时训练一个效果良好的模型需要高性能的计算资源（**GPU**）和相当的时间。其次，在训练数据范围之外的数据上，已训练模型的表现就会很差，这是因为机器并没有理解这个问题，所以不能在没有训练过的数据上进行泛化。

我们很难看到训练过的模型的内在机制，手动调参也相当困难，因为深度学习模型里面有数百万个参数——每个参数在训练过程中都需要调整。从某种意义上来说，深度学习模型就是一个黑匣子。

传统的计算机视觉具有充分的透明度，这能够使你对解决方案能否在训练环境之外运行做一个更好的评估和判断。你可以更容易地了解算法中存在的问题，弄清楚什么地方需要调整。

就我们这次的项目来说，在效果上，两种方法的都能以非常好的效果完成任务。但从时间上来说，利用传统方法在 **Inpainting** 步骤所花时间较长，通常每张图片需要 3-5 分钟甚至更长的时间，由于这种缺陷，这种方法可能就没法用于需要实时处理的现实情境当中。要加快算法的运行效率，可以从填充的模式入手，因为当前算法是在带填充区域边缘一个像素一个像素地填充，那么我们可以尝试在图像边缘一圈一圈的填充，虽然这样可能会牺牲一些图像填充的效果，但可以加快运行速度。

对于深度学习方法来说，虽然训练一个模型需要较长的时间，但训练出的

模型可以对图片较快的进行处理和输出。

总的来说，两种方法有利有弊。深度学习只是计算机视觉的一种工具，并不是万能药。不要因为深度学习现在较为流行就只使用深度学习。传统的计算机视觉技术仍然非常重要，它可以为你节省很多时间，并减少许多不必要的麻烦。

参考文献

- [1] Antonio Criminisi, Patrick Perez, and Kentaro Toyama. Region filling and object removal by exemplar-based image inpainting. *IEEE Transactions on Image Processing*, 13(9):1200–1212, 2004.
- [2] Kaiming He, Xiangyu Zhang, Shaoqing Ren, and Jian Sun. Deep residual learning for image recognition. In *Proceedings of the IEEE conference on computer vision and pattern recognition*, pages 770–778, 2016.
- [3] Kai Zhang, Wangmeng Zuo, Yunjin Chen, Deyu Meng, and Lei Zhang. Beyond a gaussian denoiser: Residual learning of deep cnn for image denoising. *IEEE Transactions on Image Processing*, 26(7):3142–3155, 2017.