

UNet

参考文献:

- 【论文】U-Net: Convolutional Networks for Biomedical Image Segmentation (<https://arxiv.org/abs/1505.04597>)
- 图像分割必备知识点 | Unet详解 理论 + 代码: <https://zhuanlan.zhihu.com/p/313283141>

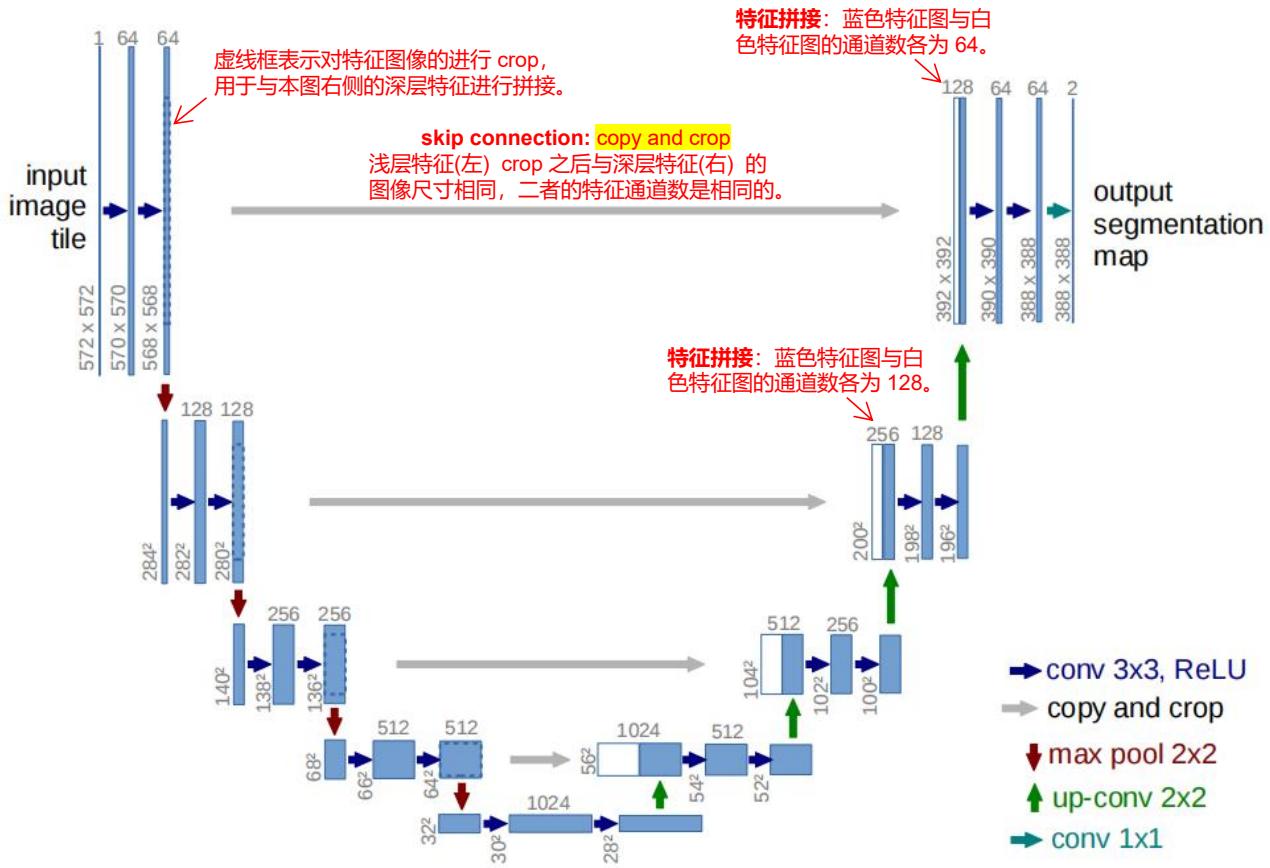


Fig. 1. U-net architecture (example for 32x32 pixels in the lowest resolution). Each blue box corresponds to a multi-channel feature map. The number of channels is denoted on top of the box. The x-y-size is provided at the lower left edge of the box. White boxes represent copied feature maps. The arrows denote the different operations.

Unet 网络非常的简单，前半部分就是特征提取，后半部分是上采样。在一些文献中把这种结构叫做**编码器-解码器**结构，由于网络的整体结构是一个大些的英文字母U，所以叫做U-net。

- Encoder: 左半部分由两个 3x3 的卷积层 (RELU) 再加上一个 2x2 的 max pooling 层组成一个下采样的模块；
- Decoder: 右半部分由一个上采样的卷积层 (去卷积层) + 特征拼接 concat + 两个 3x3 的卷积层 (ReLU) 反复构成；

在当时，Unet 相比更早提出的 FCN 网络，使用**拼接**来作为**特征图融合**的方式。

- FCN 是通过特征图对应像素值的**相加**来融合特征的；
- U-net 通过通道数的**拼接**，这样可以形成更厚的特征，当然这样会更佳消耗显存；

Unet 的好处是：

- 网络层越深得到的特征图，有着**更大的感受野**（语义，全局特征）；浅层卷积关注**纹理特征**（边界，局部特征），深层网络关注本质的那种特征，所以深层浅层特征都是有各自的意义的。
- 通过反卷积得到的更大的尺寸的特征图是缺少原有的边缘信息的，毕竟每一次下采样提炼特征的同时，也必然会损失一些边缘特征，而失去的特征并不能从上采样中找回，因此**通过特征的拼接，来实现边缘特征的一个找回**。

医疗影像的特点：

- 医疗影像语义较为简单、结构固定。因此语义信息相比自动驾驶等较为单一，因此并不需要去筛选过滤无用的信息。
医疗影像的所有特征都很重要，因此低级特征和高级语义特征都很重要，所以U型结构中的这种 skip connection 结构（特征拼接）更好派上用场。
- 医学影像的数据较少，获取难度大，数据量可能只有几百甚至不到100，因此如果使用大型的网络例如 DeepLabv3+ 等模型，很容易过拟合。大型网络的优点是更强的图像表述能力，而较为简单、数量少的医学影像并没有那么多的内容需要表述，因此也有人发现在小数量级中，分割的SOTA模型与轻量的Unet并没有太大的优势差别。
- 医学影像往往是多模态的。比方说ISLES脑梗竞赛中，官方提供了CBF, MTT, CBV等多中模态的数据。因此医学影像任务中，往往需要自己设计网络去提取不同的模态特征，因此轻量结构简单的 Unet 可以有更大的操作空间。

大多数医疗影像语义分割任务都会首先用 Unet 作为 baseline。

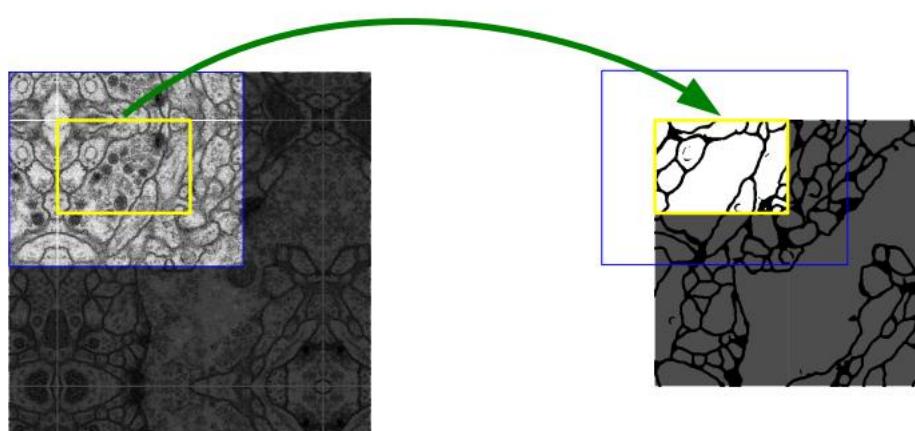


Fig. 2. Overlap-tile strategy for seamless segmentation of arbitrary large images (here segmentation of neuronal structures in EM stacks). Prediction of the segmentation in the yellow area, requires image data within the blue area as input. Missing input data is extrapolated by mirroring

语义分割网络 U-Net 详解

<https://zhuanlan.zhihu.com/p/79204199>

Unet 发表于 2015 年，属于 FCN 的一种变体。Unet 的初衷是为了解决生物医学图像方面的问题，由于效果确实很好后来也被广泛的应用在语义分割的各个方向，比如卫星图像分割，工业瑕疵检测等。

该作者另一篇关于 FCN 的文章：

- FCN 全卷积网络论文阅读及代码实现：<https://zhuanlan.zhihu.com/p/77201674>

Unet 跟 FCN 都是 **Encoder-Decoder** 结构，结构简单但很有效。Encoder 负责特征提取，你可以将自己熟悉的各种特征提取网络放在这个位置。

由于在医学方面，样本收集较为困难，作者为了解决这个问题，应用了**图像增强**的方法，在数据集有限的情况下获得了不错的精度。

为什么是 5 层而不是 4 层或者 6 层，emmm，这应该去问作者本人，可能对于当时作者拿到的数据集来说，这个层数的表现更好，但不代表所有的数据集这个结构都适合。**我们该多关注这种 Encoder-Decoder 的设计思想，具体实现则应该因数据集而异。**

Encoder

由**卷积**操作和**下采样**操作组成：

- 文中所用的卷积结构统一为 3x3 的卷积核，padding 为 0，striding 为 1。没有 padding 所以每次卷积之后 feature map 的 H 和 W 变小了，在 **skip-connection** 时要注意 feature map 的维度(其实也可以将 padding 设置为 1 避免维度不对应问题)。
- 上述的两次卷积之后是一个 stride 为 2 的 **max pooling**，输出大小变为 $1/2 * (H, W)$ 。

上面的步骤重复 5 次，最后一次没有 max-pooling (即5次双卷积、4次 maxpooling 下采样)，直接将得到的 feature map 送入 Decoder。

pytorch 代码：

```
nn.Sequential(nn.Conv2d(in_channels, out_channels, 3),  
             nn.BatchNorm2d(out_channels),  
             nn.ReLU(inplace=True))
```

```
nn.MaxPool2d(kernel_size=2, stride=2)
```

Decoder

feature map 经过 Decoder 恢复原始分辨率，该过程除了卷积比较关键的步骤就是 **upsampling** 与 **skip-connection**。

Upsampling 上采样常用的方式有两种：

1. FCN 中介绍的**反卷积**；
2. **插值**。 (下一页详细介绍 bilinear 差值)

CNN 网络要想获得好效果，skip-connection 基本必不可少。Unet 中这一关键步骤融合了**浅层特征的位置信息**与**深层特征的语义信息**。

pytorch 代码：

```
nn.Upsample(scale_factor=2, mode='bilinear')  
  
torch.cat([low_layer_features, deep_layer_features], dim=1)
```

这里需要注意的是，FCN 中深层信息与浅层信息融合是通过**对应像素相加**的方式，而 Unet 是通过**拼接**的方式。

那么这两者有什么区别呢？

- 其实在 ResNet 与 DenseNet 中也有一样的区别，Resnet 使用了对应值相加，DenseNet 使用了拼接。
- 个人理解：
 - 在相加的方式下，feature map 的维度没有变化，但每个维度都包含了更多特征，对于普通的**分类任务**这种不需要从 feature map 复原到原始分辨率的任务来说，这（相加）是一个高效的选择；
 - 而拼接则保留了更多的维度/位置 信息，这使得后面的 layer 可以在浅层特征与深层特征自由选择，这对**语义分割任务**来说更有优势。

论文采用不 padding 的意义：

padding会影响感受野，应该是不想压缩有效数据占比吧。 (来自知乎网友回答)

总结：

这里介绍文中使用的插值方式。在插值实现方式中，**bilinear 双线性插值**的综合表现较好也较为常见。双线性插值的计算过程没有需要学习的参数，实际就是套公式。

这里举个例子方便大家理解（例子介绍的是参数 align_corners 为 False 的情况）。

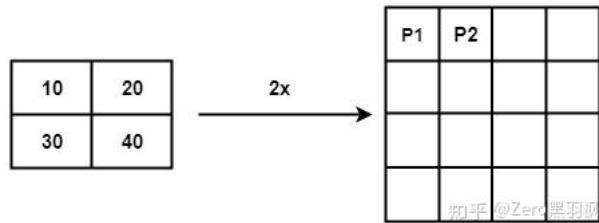
```
In [1]: import torch  
import numpy as np  
  
src = torch.Tensor(np.asarray([[[[10, 20], [30, 40]]]]))  
print("src: ")  
print(src)  
upsample = torch.nn.Upsample(scale_factor=2, mode="bilinear", align_corners=False)  
print("dst: ")  
print(upsample(src))
```

src:
tensor([[[[10., 20.],
 [30., 40.]]]])
dst:
tensor([[[[10.0000, 12.5000, 17.5000, 20.0000],
 [15.0000, 17.5000, 22.5000, 25.0000],
 [25.0000, 27.5000, 32.5000, 35.0000],
 [30.0000, 32.5000, 37.5000, 40.0000]]]])

知乎 @Zero黑羽枫

例子中是将一个 2x2 的矩阵通过插值的方式得到 4x4 的矩阵，那么将 2x2 的矩阵称为源矩阵，4x4 的矩阵称为目标矩阵。双线性插值中，目标点的值是由离他最近的 4 个点的值计算得到的，我们首先介绍如何找到目标点周围的 4 个点，以 P2 为例。

第一个公式，目标矩阵到源矩阵的坐标映射：



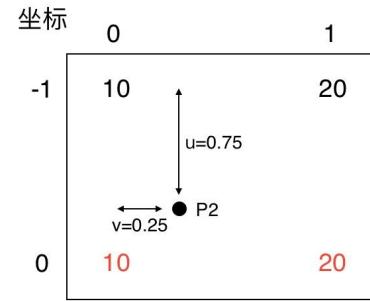
$$X_{src} = (X_{dst} + 0.5) * \left(\frac{Width_{src}}{Width_{dst}} \right) - 0.5$$

$$Y_{src} = (Y_{dst} + 0.5) * \left(\frac{Height_{src}}{Height_{dst}} \right) - 0.5$$

为了找到那 4 个点，首先要找到目标点在源矩阵中的**相对位置**，上面的公式就是用来算这个的。P2 在目标矩阵中的坐标是 (0, 1)，对应到源矩阵中的坐标就是 (-0.25, 0.25)。坐标里面居然有小数跟负数，不急我们一个一个来处理。

我们知道双线性插值是从坐标周围的 4 个点来计算该坐标的值，(-0.25, 0.25) 这个点周围的 4 个点是(-1, 0), (-1, 1), (0, 0), (0, 1)。为了找到负数坐标点，我们将源矩阵扩展为下面的形式，中间红色的部分为源矩阵。

10	10	20	20
10	10	20	20
30	30	40	40
30	30	40	40



知乎 @Zero黑羽枫

我们规定 $f(i, j)$ 表示 (i, j) 坐标点处的像素值，对于计算出来的对应的坐标，我们统一写成 $(i+u, j+v)$ 的形式。那么这时 $i=-1$, $u=0.75$, $j=0$, $v=0.25$ 。把这 4 个点单独画出来，可以看到目标点 P2 对应到源矩阵中的相对位置。

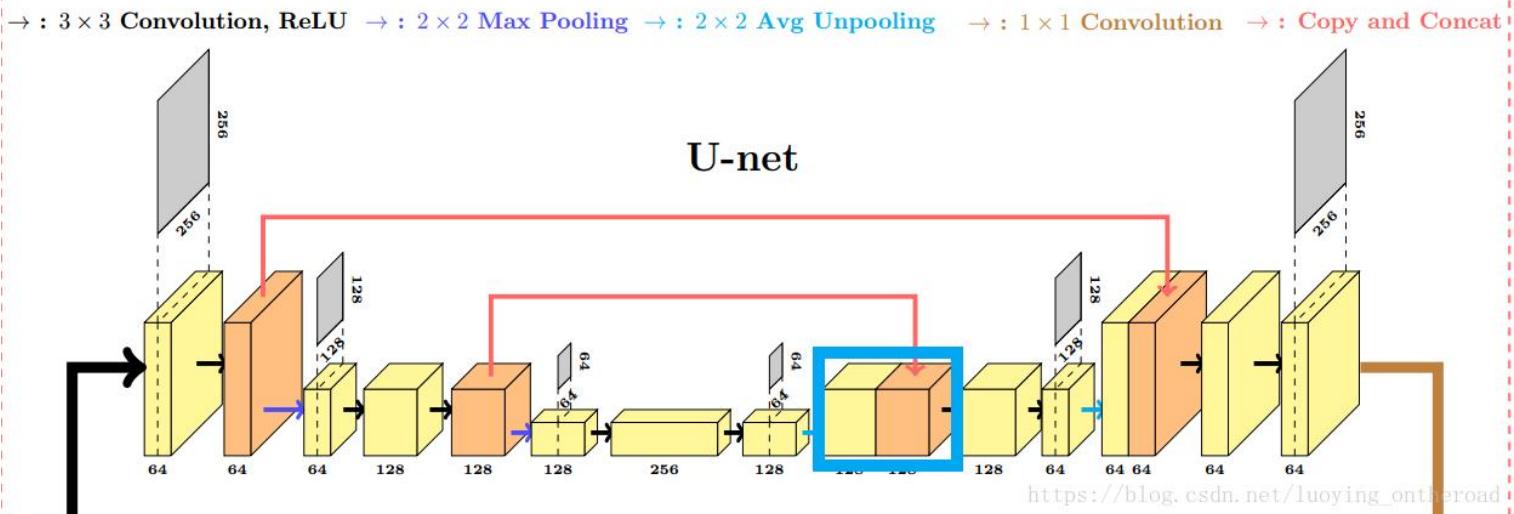
$$f(i+u, j+v) = (1-u)(1-v)f(i, j) + (1-u)v f(i, j+1) + u(1-v)f(i+1, j) + u v f(i+1, j+1)$$

目标点的像素值就是周围 4 个点像素值的加权和，明显可以看出离得近的权值比较大例如 (0, 0) 点的权值就是 $0.75*0.75$ ，离得远的如 (-1, 1) 权值就比较小，为 $0.25*0.25$ ，这也比较符合常理吧。把值带入计算就可以得到 P2 点的值了，结果是 12.5 与代码结果吻合，nice。

总结：

unet模型及代码解析

https://blog.csdn.net/qq_43537701/article/details/121177321



UNet 与 FCN 的比较

1. 编解码结构：它们的结构都用了一个比较经典的思路，也就是 编码和解码（encoder-decoder）结构，该结构早在 2006 年就被 Hinton 提出来发表在了 nature 上。当时这个 encoder-decoder 结构提出的主要作用并不是分割，而是压缩图像和去噪声。和 FCN 相比，U-Net 的第一个特点是完全对称，也就是左边和右边是很类似的，而 FCN 的解码器部分相对简单，只用了一个反卷积的操作，之后并没有跟上卷积结构。

2. 全卷积结构：UNet 和 FCN 一样，是全卷积形式，没有全连接层(即没有固定图的尺寸)，不像全连接层输入是提前固定好的，所以容易适应很多输入尺寸大小。

3. 跳跃结构 (skip-connection)，即特征融合操作：

- UNet 相比 FCN，跳跃结构更多，更彻底，每一层下采样都与后面每一次上采样对应，一个经验的解释（大量实验）就是跳级连接能够保证特征更加精细。
- UNet 是拼接操作，而 FCN 是加操作。

对高层语义特征与底层空间信息的理解：

越底层浅层的特征蕴含的空间信息（分割定位特征）更多，语义特征（就是类别判断特征，像素点可以分到哪一个类别中去）更少，越高级深层的特征蕴含的空间信息更少，语义特征更多。

- 底层浅层特征图片更偏向于组成图像的基本单元，如点，线，边缘轮廓。（low-level 特征）
 - 高层深层的特征就更抽象，更近似于表示的是图像的语义信息。（high-level 特征）
- 结合论文：Visualizing and Understanding Convolutional Networks
- 论文网址：https://blog.csdn.net/qq_43537701/article/details/121177321

小样本分割经典网络：U-Net解析

<https://zhuanlan.zhihu.com/p/370931792>

University of Freiburg, Germany, 德国弗莱堡大学：

• <https://link.zhihu.com/?target=http%3A//lmb.informatik.uni-freiburg.de/>

完整版的实现和已经训练好的网络参数在这里：

• <https://link.zhihu.com/?target=http%3A//lmb.informatik.uni-freiburg.de/people/ronneber/u-net>

如何解决生物医学上的大图片所带来的显存需求急剧升高问题？

- 网络中没有任何全连接层，只对特征图本身进行了卷积。这样操作就使最后得到的分割遮罩就对应于原图的每个像素，输入图片中就包含所有的上下文纹理信息。
- 这样的只对特征图本身进行卷积的策略就可以对任何的大图片进行无缝分割，如图2所示，进行 **overlap-tile** 策略。
 - 对于图片中心的像素，我们使用其周围区域来分割它。
 - 对于图片边缘的像素，它没有周围像素，所以我们把图片内像素镜像到图片外，作为边缘像素的周围像素。
 - 这种 tiling strategy 很重要，因为它的存在，我们就可以把网络应用于大图片了。一次分割大图片的一个区域即可。训练时还可以不受GPU显存的限制。

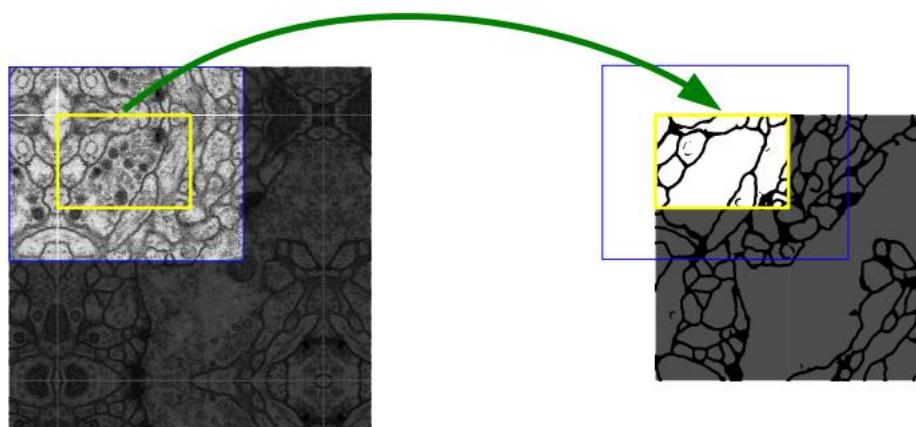


Fig. 2. Overlap-tile strategy for seamless segmentation of arbitrary large images (here segmentation of neuronal structures in EM stacks). Prediction of the segmentation in the yellow area, requires image data within the blue area as input. Missing input data is extrapolated by mirroring

如何解决生物医学图像的数据非常少的问题？

我们使用了**大量的数据增强**，对有限的数据进行了 **elastic deformations** (弹性变形)。这一点在生物医学分割上也非常重要，这种形变非常适合生物组织，形变后也是栩栩如生的。

具体数据增强的参数在此文的无导师特征学习部分有介绍：

[Discriminative un-supervised feature learning with convolutional neural networks.](#)

如何解决待分割物体占图片像素比重很大问题？

关于细胞分割的另外一个挑战是需要分割的细胞紧密相连，相互 touching 的。前景部分很多，背景部分很少，所以，我们提出使用了一个加权 loss，使细胞之间的缝隙有较大的训练权重。

如图3所示，图3 d图的红色部分是权重比较大的部分。

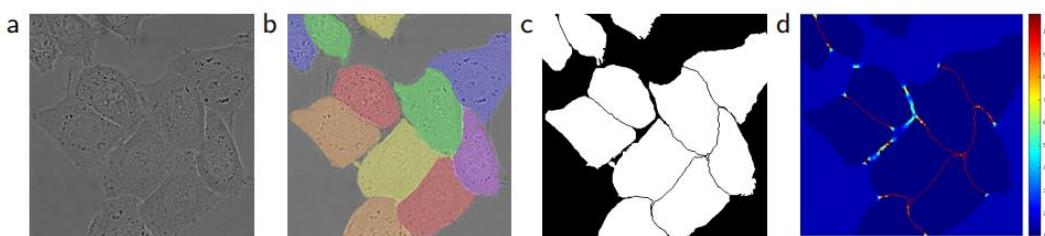
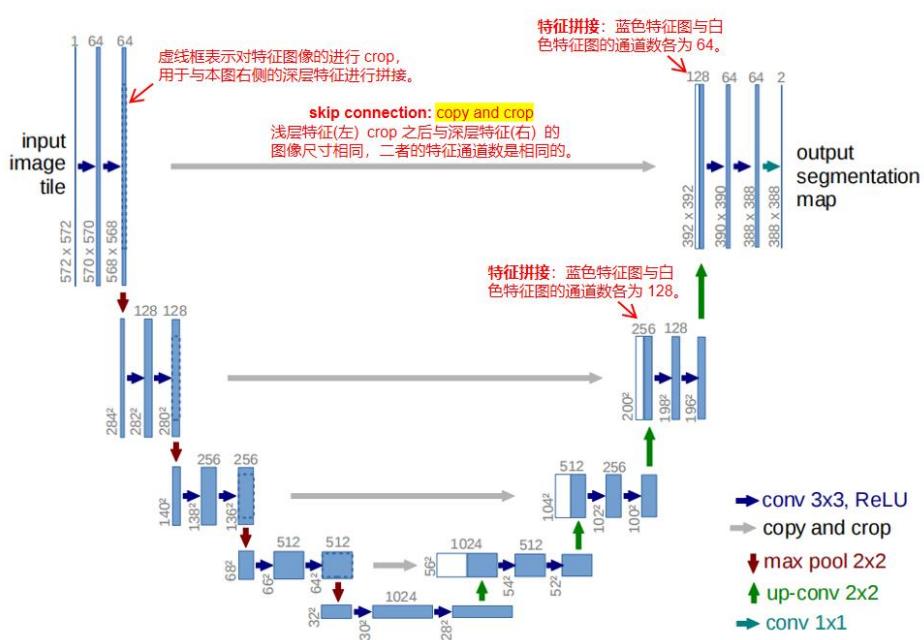


Fig. 3. HeLa cells on glass recorded with DIC (differential interference contrast) microscopy. (a) raw image. (b) overlay with ground truth segmentation. Different colors indicate different instances of the HeLa cells. (c) generated segmentation mask (white: foreground, black: background). (d) map with a pixel-wise loss weight to force the network to learn the border pixels.

解决了以上三个问题的网络可以应对各种各样的生物医学分割问题。

网络结构 (整个网络一共有23个卷积层)

- 下采样阶段是典型的卷积神经网络结构。包含重复的两个 3×3 卷积操作（无padded卷积）。每个卷积后面跟着一个 rectified linear unit (ReLU) 和一个 2×2 的池化层，stride=2 实现下采样功能。在每一次下采样时，我们时 channels 数加倍。
- 上采样阶段的每一步结构是一样的，都包含一个上采样得到的 feature map，然后对这个 feature map 进行 2×2 up-convolution, 它使 feature map 大小增加，但是 channels 减半，然后在 channels 维度上与 crop 过的下采样阶段的 feature map 进行拼接。拼接后进行两个 3×3 的卷积，每个卷积都跟着一个ReLU。这个过程中的 cropping 是必须的，因为随着无 padded convolution 的进行，feature map 的大小在减少。
- 最后一层的 1×1 convolution是为了将64channel映射为类别数。



训练过程

- 训练过程使用了caffe中已经实现的随机梯度下降法
- 为了尽量减少重叠，尽最大程度的利用GPU的显存，使用了比输出图片大的输入粘贴图，使用了大的batch size，从而降低单张图片的batch。
- momentum=0.99，非常接近于1了，非常高。这会让网络朝着根据之前见过的大部分数据的方向去优化。
- 损失函数（能量函数）是在最后一层 feature map 上的像素级的 soft-max，然后使用 cross entropy loss function。

The energy function is computed by a pixel-wise soft-max over the final feature map combined with the cross entropy loss function. The soft-max is defined as $p_k(\mathbf{x}) = \exp(a_k(\mathbf{x}))/\left(\sum_{k'=1}^K \exp(a_{k'}(\mathbf{x}))\right)$ where $a_k(\mathbf{x})$ denotes the activation in feature channel k at the pixel position $\mathbf{x} \in \Omega$ with $\Omega \subset \mathbb{Z}^2$. K is the number of classes and $p_k(\mathbf{x})$ is the approximated maximum-function. I.e. $p_k(\mathbf{x}) \approx 1$ for the k that has the maximum activation $a_k(\mathbf{x})$ and $p_k(\mathbf{x}) \approx 0$ for all other k . The cross entropy then penalizes at each position the deviation of $p_{\ell(\mathbf{x})}(\mathbf{x})$ from 1 using

$$E = \sum_{\mathbf{x} \in \Omega} w(\mathbf{x}) \log(p_{\ell(\mathbf{x})}(\mathbf{x})) \quad (1)$$

where $\ell : \Omega \rightarrow \{1, \dots, K\}$ is the true label of each pixel and $w : \Omega \rightarrow \mathbb{R}$ is a weight map that we introduced to give some pixels more importance in the training.

We pre-compute the weight map for each ground truth segmentation to compensate the different frequency of pixels from a certain class in the training data set, and to force the network to learn the small separation borders that we introduce between touching cells (See Figure 3c and d).

The separation border is computed using morphological operations. The weight map is then computed as

$$w(\mathbf{x}) = w_c(\mathbf{x}) + w_0 \cdot \exp\left(-\frac{(d_1(\mathbf{x}) + d_2(\mathbf{x}))^2}{2\sigma^2}\right) \quad (2)$$

where $w_c : \Omega \rightarrow \mathbb{R}$ is the weight map to balance the class frequencies, $d_1 : \Omega \rightarrow \mathbb{R}$ denotes the distance to the border of the nearest cell and $d_2 : \Omega \rightarrow \mathbb{R}$ the distance to the border of the second nearest cell. In our experiments we set $w_0 = 10$ and $\sigma \approx 5$ pixels.

ℓ ：分割的类别。

w ：每个像素对应的权重【前面说错过缝隙位置要加大权重，因为缝隙位置少】。

每个像素权重需要预先计算出来，通过分割的gt来计算。

每个像素加权主要是为了弥补像素种，不同类别的数据不平衡问题。强迫网络学习相互紧贴的细胞之间的像素类别。

权重公式主要从形态上考虑，如公式(2)所示。

- d_1 、 d_2 ：分别表示当前像素距离细胞边缘的第一近距离和第二近距离。
- w_c ：是类别平衡矩阵，每个像素对应一个值。

Unet相关介绍 (https://blog.csdn.net/weixin_41630455/article/details/116635526) 中说：

我的理解是 w_c 就是一个数值，只不过 c 会有不同取值， c 的取值和训练集的语义类别相关。如果这个语义类别在总像素中占比小那么 w_c 这个值就大，反之则小。 w_c 是用来平衡类别的差异的权重，而公式的后半段是用来加强边界像素误差的权重。

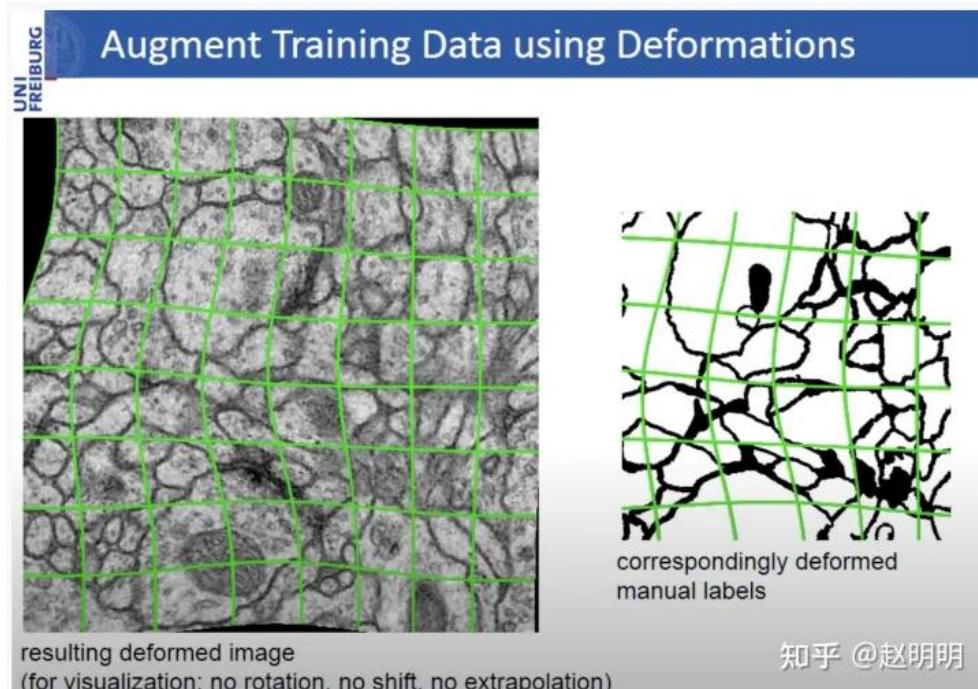
总结：

U-Net是如何数据增强的？

若要网络性能好，当数据量分太少时，数据增强是必须的。

显微图片的话，我们主要通过**平移、旋转**来增强。

但是**随机的弹性形变**对于训练一个分割网络来说十分必要。我们使用随机移位向量在 3×3 的网络上实现平滑的移位。向量来自于 10 像素标准差的Gaussian分布，每个像素的值用双三次线性插值来计算。



弹性变换后的图像

Unet相关介绍

https://blog.csdn.net/weixin_41630455/article/details/116635526

Unet 是在 FCN 基础上提出的一种应用于医学影响的分割网络。医学影像的特点是：

- 1、数据集小。
- 2、单张图片大。

由于以上医疗影像的特点，我们无法直接用 FCN 进行分割学习。一个德国团队提出了 Unet 网络设计在，做到了仅仅用30张医疗影响的数据集就取得了相当不错的效果。

医疗影像数据样本有一个问题就是数据不均衡的问题。我们发现医疗影像的背景像素在图中的占比是非常小的。像这样的数据不均衡问题直接在 loss 上做文章。结合论文看一下具体的做法，利用了一个加权的loss，其中背景 loss 分配了一个大的权重。

Another challenge in many cell segmentation tasks is the separation of touching objects of the same class; see Figure 3. To this end, we propose the use of a weighted loss, where the separating background labels between touching cells obtain a large weight in the loss function.

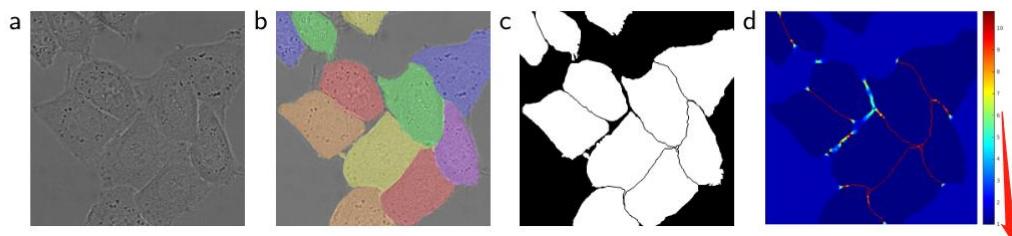


Fig. 3. HeLa cells on glass recorded with DIC (differential interference contrast) microscopy. (a) raw image. (b) overlay with ground truth segmentation. Different colors indicate different instances of the HeLa cells. (c) generated segmentation mask (white: foreground, black: background). (d) map with a pixel-wise loss weight to force the network to learn the border pixels.

https://blog.csdn.net/weixin_41630455

我认为 Unet 与 FCN 主要有以下几点差异：

- 1、使用了 **overlap-tile** 策略。它是对一张大的医疗影响先分成一小块一小块进行分割训练（388*388大小的子图）。最终将这一小块一小块的分割图拼接起来。这个与 fcn 相比，fcn 的尺寸是直接拿原图的尺寸，而医疗原图的尺寸是很大的，训练起来 feature map 尺寸就很大计算要求很高。另外这里分割成一个个小图直接解决了医疗影响数据集图片少的问题。为了结果能够无缝进行拼接，每次下采样 maxpooling 时 x, y 方向上的尺寸都需要是偶数被2给整除。

To allow a seamless tiling of the output segmentation map (see Figure 2), it is important to select the input tile size such that all 2x2 max-pooling operations are applied to a layer with an even x- and y-size.

- 2、上图写的 input 后面有个 tile。网络输入的是 572*572 尺寸的，最后结果是 388*388 的尺寸的。这里 572*572 可以认为是 388*388 的 padding 出来的。和 fcn 不同的是，fcn 比较简单粗暴直接 padding 了100 的0。这里认为直接 padding 0 对于边缘信息处理是不好的。**Unet 是拿原图边缘的内容进行填充**。（如果 388*388 子图正好在边缘位置没法用原图 padding，则使用了**镜像 padding** 的方法，直接对应位置做了个镜像填充进去）。

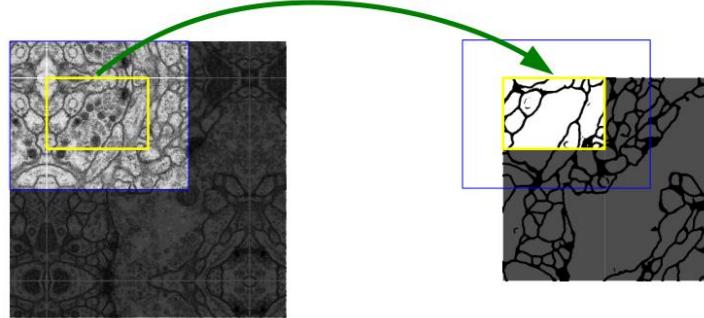


Fig. 2. Overlap-tile strategy for seamless segmentation of arbitrary large images (here segmentation of neuronal structures in EM stacks). Prediction of the segmentation in the yellow area, requires image data within the blue area as input. Missing input data is extrapolated by mirroring

https://blog.csdn.net/weizin_41630455

- 3、**特征融合程度**区分。fcn 最多融合到8s。而 Unet 对所有上采样的特征图与原图进行了融合，融合了更多的尺度。且 Unet 的模型结构是左右对称的。
- 4、**特征融合方式**不同。fcn 融合的时候采用了+ 法，将特征图上的 element 进行相加。而 Unet 用了 concat 的方法在channel 维度上进行了堆叠。（Unet由于 overlap 的策略网络计算更高效了，而 concat 增加了 channel 对应增加的计算量在这里是可以接受的）。
- 5、关于 Unet 训练的**数据增强方式**。可以使用图像的**弹性变化**进行增强。这个也是医疗影像数据的一个特性。