**超分辨率技术（Super-Resolution）**

超分辨率技术（Super-Resolution, SR）是指从观测到的低分辨率图像重建出相应的高分辨率图像，在监控设备、卫星图像和医学影像等领域都有重要的应用价值。

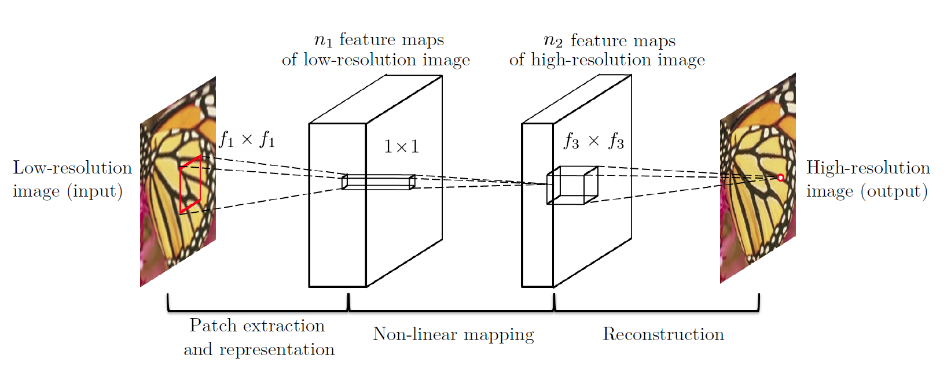
本文针对端到端的基于深度学习的单张图像超分辨率方法(Single Image Super-Resolution, SISR)，总结一下从SRCNN到EDSR的发展历程。(排列顺序大致按论文中给出的4倍上采样结果的峰值信噪比(Peak Signal to Noise Ratio, PSNR)从低到高排列)

**1. SRCNN**

([Learning a Deep Convolutional Network for Image Super-Resolution](http://personal.ie.cuhk.edu.hk/~ccloy/files/eccv_2014_deepresolution.pdf), ECCV2014)

code: <http://mmlab.ie.cuhk.edu.hk/projects/SRCNN.html>

SRCNN是深度学习用在超分辨率重建上的开山之作。SRCNN的网络结构非常简单，仅仅用了三个卷积层，网络结构如下图所示。



SRCNN首先使用双三次(bicubic)插值将低分辨率图像放大成目标尺寸，接着通过三层卷积网络拟合非线性映射，最后输出高分辨率图像结果。本文中，作者将三层卷积的结构解释成三个步骤：图像块的提取和特征表示，特征非线性映射和最终的重建。

三个卷积层使用的卷积核的大小分为为9x9,，1x1和5x5，前两个的输出特征个数分别为64和32。用Timofte数据集（包含91幅图像）和ImageNet大数据集进行训练。使用均方误差(Mean Squared Error, MSE)作为损失函数，有利于获得较高的PSNR。

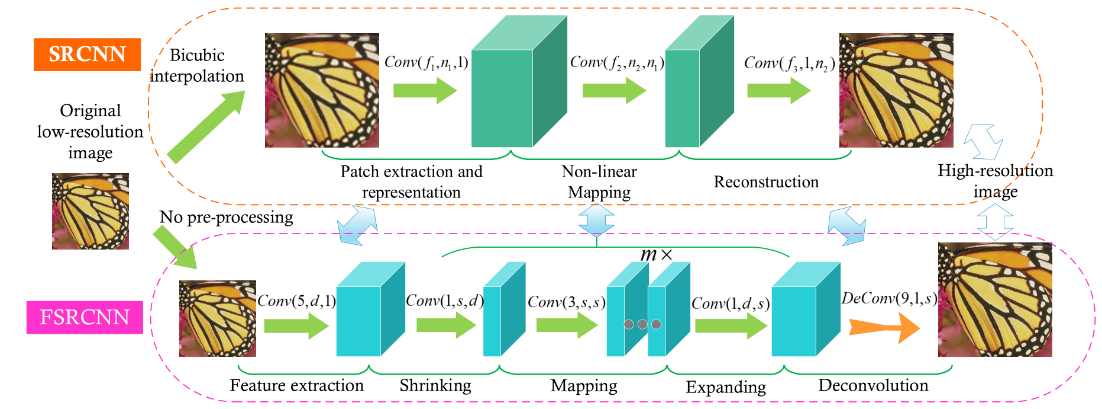
**2. FSRCNN**

([Accelerating the Super-Resolution Convolutional Neural Network](https://arxiv.org/abs/1608.00367), ECCV2016)

code: <http://mmlab.ie.cuhk.edu.hk/projects/FSRCNN.html>

FSRCNN与SRCNN都是香港中文大学Dong Chao， Xiaoou Tang等人的工作。FSRCNN是对之前SRCNN的改进，主要在三个方面：一是在最后使用了一个反卷积层放大尺寸，因此可以直接将原始的低分辨率图像输入到网络中，而不是像之前SRCNN那样需要先通过bicubic方法放大尺寸。二是改变特征维数，使用更小的卷积核和使用更多的映射层。三是可以共享其中的映射层，如果需要训练不同上采样倍率的模型，只需要fine-tuning最后的反卷积层。

由于FSRCNN不需要在网络外部进行放大图片尺寸的操作，同时通过添加收缩层和扩张层，将一个大层用一些小层来代替，因此FSRCNN与SRCNN相比有较大的速度提升。FSRCNN在训练时也可以只fine-tuning最后的反卷积层，因此训练速度也更快。FSRCNN与SCRNN的结构对比如下图所示。



FSRCNN可以分为五个部分。特征提取：SRCNN中针对的是插值后的低分辨率图像，选取的核大小为9×9，这里直接是对原始的低分辨率图像进行操作，因此可以选小一点，设置为5×5。收缩：通过应用1×1的卷积核进行降维，减少网络的参数，降低计算复杂度。非线性映射：感受野大，能够表现的更好。SRCNN中，采用的是5×5的卷积核，但是5×5的卷积核计算量会比较大。用两个串联的3×3的卷积核可以替代一个5×5的卷积核，同时两个串联的小卷积核需要的参数3×3×2=18比一个大卷积核5×5=25的参数要小。FSRCNN网络中通过m个核大小为3×3的卷积层进行串联。扩张：作者发现低维度的特征带来的重建效果不是太好，因此应用1×1的卷积核进行扩维，相当于收缩的逆过程。反卷积层：可以堪称是卷积层的逆操作，如果步长为n，那么尺寸放大n倍，实现了上采样的操作。

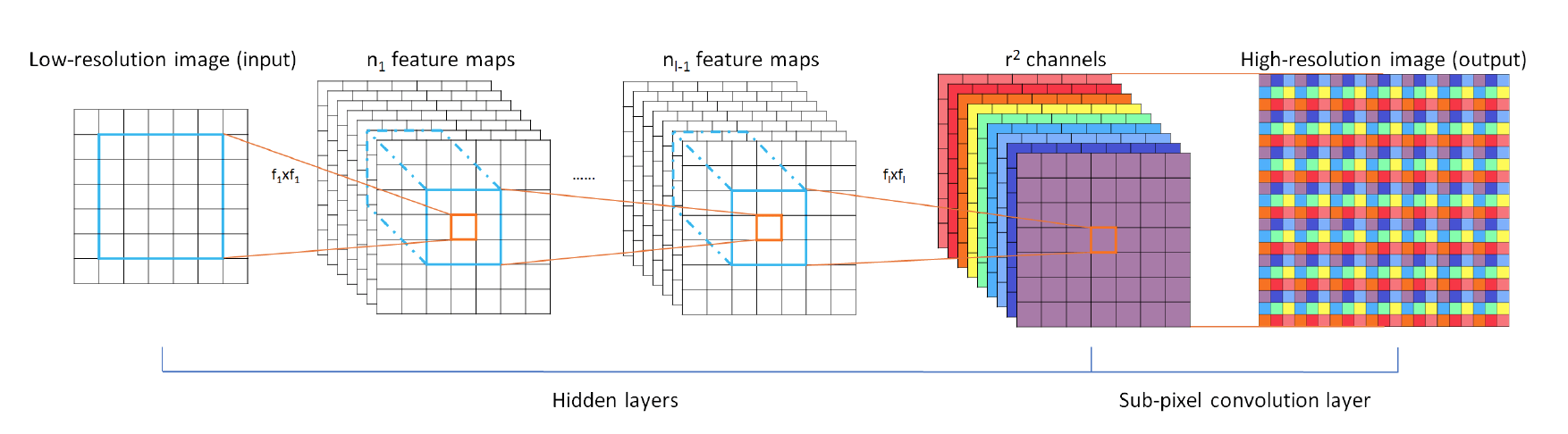
FSRCNN中激活函数采用PReLU，损失函数仍然是均方误差。对CNN来说，Set91并不足够去训练大的网络结构，FSRCNN提出general-100 + Set91进行充当训练集。并且进行数据增强，1）缩小尺寸为原来的0.9, 0.8, 0.7和0.6。2）旋转 90°，180°和270°，因此获得了数据量的提升。

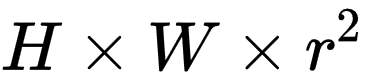
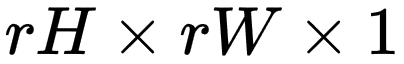
**3. ESPCN**

([Real-Time Single Image and Video Super-Resolution Using an Efficient Sub-Pixel Convolutional Neural Network](https://arxiv.org/abs/1609.05158), CVPR2016)

github(tensorflow): <https://github.com/drakelevy/ESPCN-TensorFlow>

作者在本文中介绍到，像SRCNN那样的方法，由于需要将低分辨率图像通过上采样插值得到与高分辨率图像相同大小的尺寸，再输入到网络中，这意味着要在较高的分辨率上进行卷积操作，从而增加了计算复杂度。本文提出了一种直接在低分辨率图像尺寸上提取特征，计算得到高分辨率图像的高效方法。ESPCN网络结构如下图所示。



ESPCN的核心概念是亚像素卷积层(sub-pixel convolutional layer)。网络的输入是原始低分辨率图像，通过三个卷积层以后，得到通道数为的与输入图像大小一样的特征图像。再将特征图像每个像素的个通道重新排列成一个的区域，对应高分辨率图像中一个大小的子块，从而大小为的特征图像被重新排列成的高分辨率图像。我理解的亚像素卷积层包含两个过程，一个普通的卷积层和后面的排列像素的步骤。就是说，最后一层卷积层输出的特征个数需要设置成固定值，即放大倍数r的平方，这样总的像素个数就与要得到的高分辨率图像一致，将像素进行重新排列就能得到高分辨率图。

在ESPCN网络中，图像尺寸放大过程的插值函数被隐含地包含在前面的卷积层中，可以自动学习到。由于卷积运算都是在低分辨率图像尺寸大小上进行，因此效率会较高。

训练时，可以将输入的训练数据，预处理成重新排列操作前的格式，比如将21×21的单通道图，预处理成9个通道，7×7的图，这样在训练时，就不需要做重新排列的操作。另外，ESPCN激活函数采用tanh替代了ReLU。损失函数为均方误差。

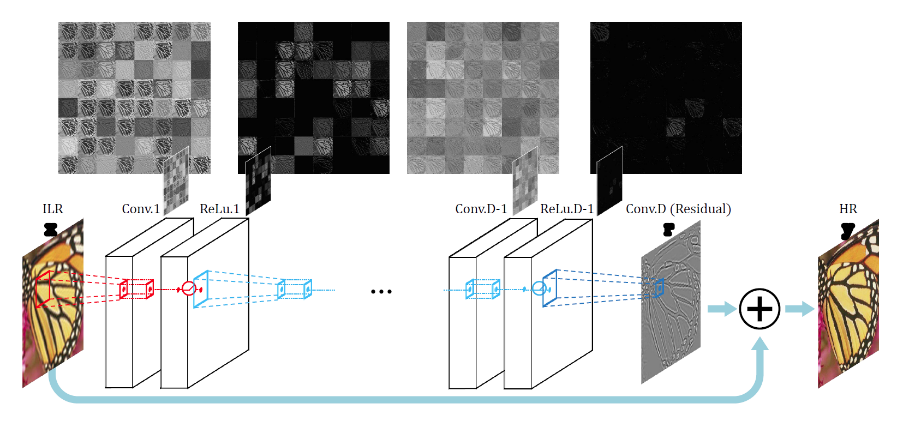
**4. VDSR**

([Accurate Image Super-Resolution Using Very Deep Convolutional Networks](https://arxiv.org/abs/1511.04587), CVPR2016)

github(tensorflow): <https://github.com/Jongchan/tensorflow-vdsr>

在介绍VDSR之前，首先想先提一下何恺明在2015年的时候提出的残差网络ResNet。ResNet的提出，解决了之前网络结构比较深时无法训练的问题，性能也得到了提升，ResNet也获得了CVPR2016的best paper。残差网络结构(residual network)被应用在了大量的工作中。

正如在VDSR论文中作者提到，输入的低分辨率图像和输出的高分辨率图像在很大程度上是相似的，也就是指**低分辨率图像携带的低频信息**与**高分辨率图像的低频信息**相近，训练时带上这部分会多花费大量的时间，实际上我们只需要学习高分辨率图像和低分辨率图像之间的**高频部分残差**即可。残差网络结构的思想特别适合用来解决超分辨率问题，可以说影响了之后的深度学习超分辨率方法。VDSR是最直接明显的学习残差的结构，其网络结构如下图所示。



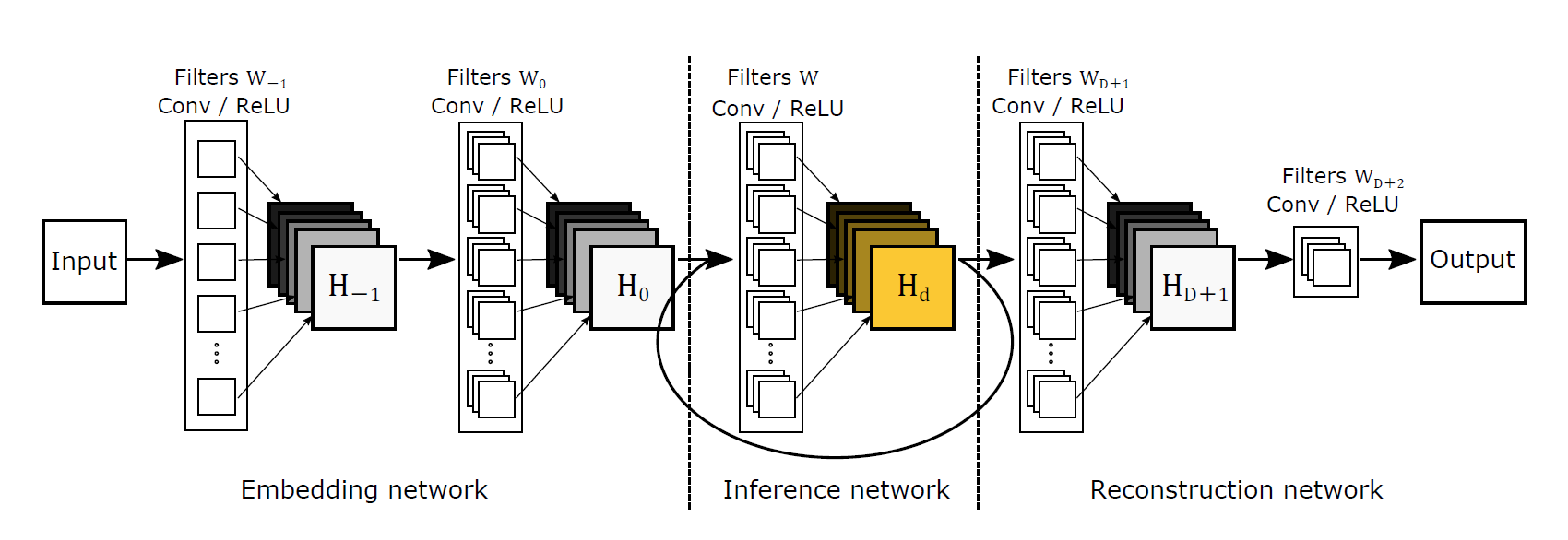
VDSR将插值后得到的变成目标尺寸的低分辨率图像作为网络的输入，再将这个图像与网络学到的残差相加得到最终的网络的输出。VDSR主要有4点贡献。1.加深了网络结构(20层)，使得越深的网络层拥有更大的感受野。文章选取3×3的卷积核，深度为D的网络拥有(2D+1)×(2D+1)的感受野。2.采用残差学习，残差图像比较稀疏，大部分值都为0或者比较小，因此收敛速度快。VDSR还应用了自适应梯度裁剪(Adjustable Gradient Clipping)，将梯度限制在某一范围，也能够加快收敛过程。3.VDSR在每次卷积前都对图像进行补0操作，这样保证了所有的特征图和最终的输出图像在尺寸上都保持一致，解决了图像通过逐步卷积会越来越小的问题。文中说实验证明补0操作对边界像素的预测结果也能够得到提升。4.VDSR将不同倍数的图像混合在一起训练，这样训练出来的一个模型就可以解决不同倍数的超分辨率问题。

**5. DRCN**

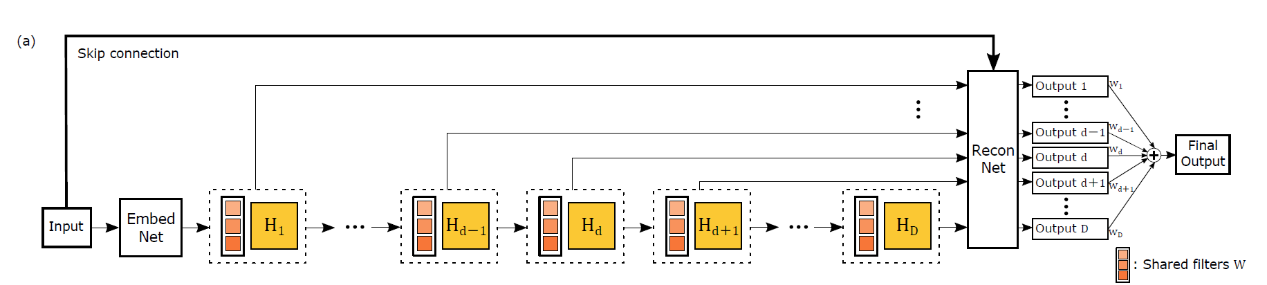
([Deeply-Recursive Convolutional Network for Image Super-Resolution](https://arxiv.org/abs/1511.04491), CVPR2016)

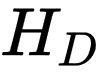
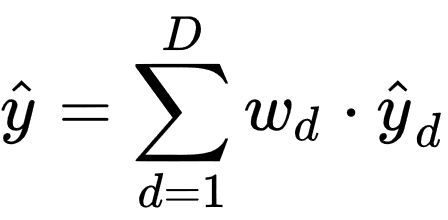
github(tensorflow): <https://github.com/jiny2001/deeply-recursive-cnn-tf>

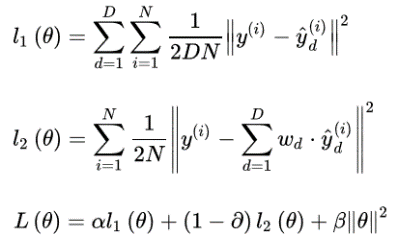
DRCN与上面的VDSR都是来自首尔国立大学计算机视觉实验室的工作，两篇论文都发表在CVPR2016上，两种方法的结果非常接近。DRCN第一次将之前已有的递归神经网络(**Recursive Neural Network**)结构应用在超分辨率处理中。同时，利用残差学习的思想(文中的跳跃连接（Skip-Connection）)，加深了网络结构(16个递归)，增加了网络感受野，提升了性能。DRCN网络结构如下图所示。



DRCN输入的是插值后的图像，分为三个模块，第一个是Embedding network，相当于特征提取，第二个是Inference network, 相当于特征的非线性映射，第三个是Reconstruction network,即从特征图像恢复最后的重建结果。其中的Inference network是一个**递归网络**，即数据循环地通过该层多次。将这个循环进行展开，**等效于使用同一组参数的多个串联的卷积层**，如下图所示。



其中的到是D个共享参数的卷积层。将这D个卷积层的每一层的结果都通过相同的Reconstruction Net，在Reconstruction Net中与输入的图像相加，得到D个输出重建结果。这些所有的结果在训练时都同时被监督，即所有的递归都被监督，作者称之为递归监督(Recursive-Supervision)，避免了梯度消失/爆炸问题。将D个递归得到的结果再加权平均：得到一个总输出。每个加权在训练的过程中也不断地更新。最终的目标函数就需要优化每一个递归层输出的误差和总输出的误差：



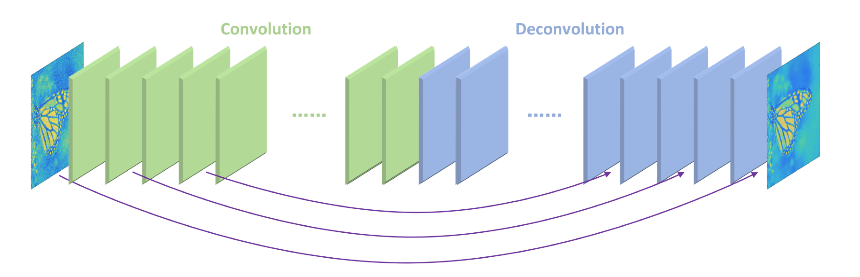
表示的是权值衰减(weight decay)。的初始值设置得比较高以使得训练过程稳定，因为训练开始的阶段递归更容易收敛。随着训练的进行，逐渐衰减来提升最终输出的性能。



**6. RED**

([Image Restoration Using Convolutional Auto-encoders with Symmetric Skip Connections](https://arxiv.org/abs/1606.08921), NIPS2016)

这篇文章提出了由对称的卷积层-反卷积层构成的网络结构，作为一个编码-解码框架，可以学习由低质图像到原始图像端到端的映射。网络结构如下图所示。



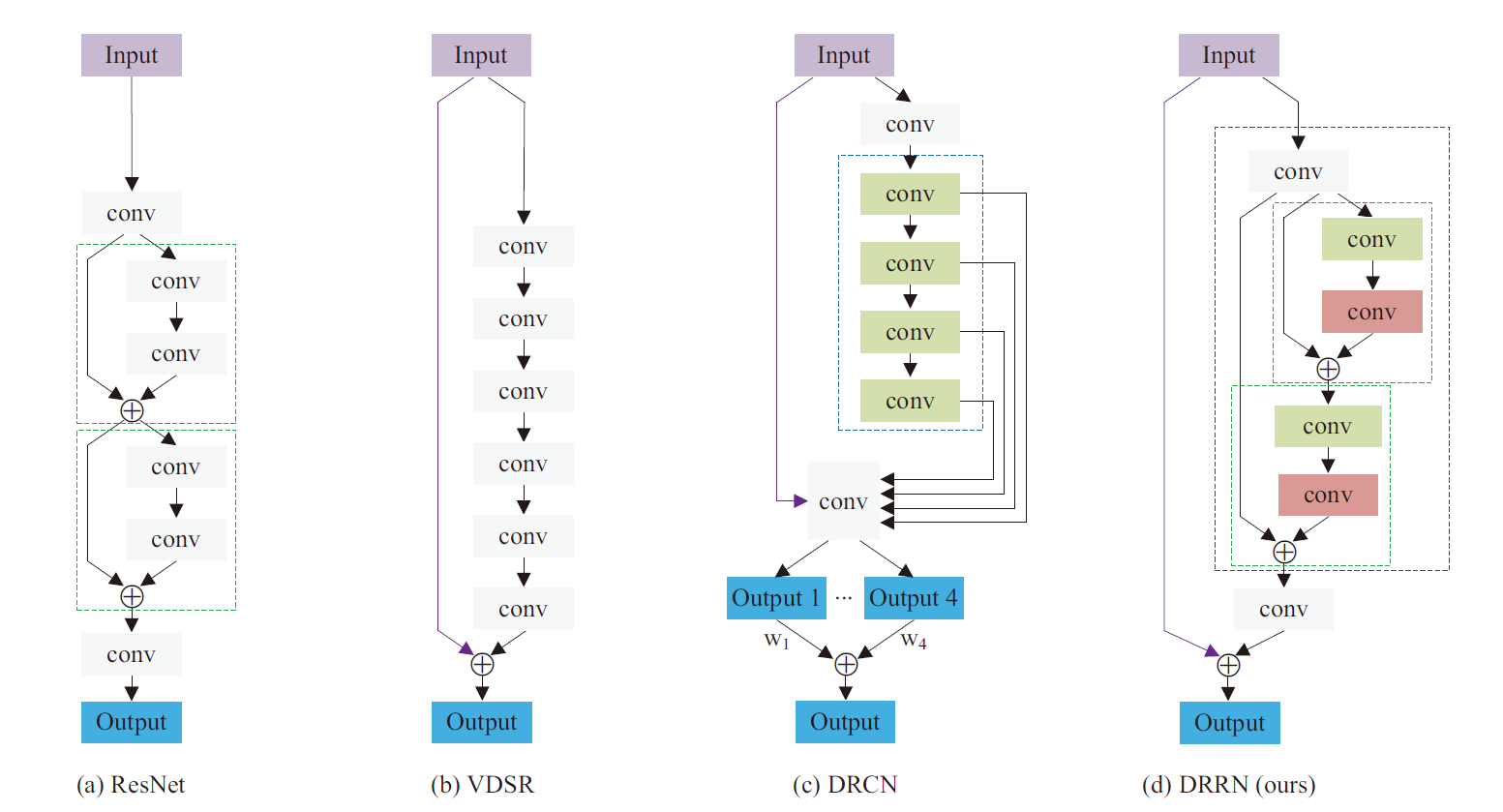
RED网络的结构是对称的，每个卷积层都有对应的反卷积层。卷积层用来获取图像的抽象内容，反卷积层用来放大特征尺寸并且恢复图像细节。卷积层将输入图像尺寸减小后，再通过反卷积层上采样变大，使得输入输出的尺寸一样。每一组镜像对应的卷积层和反卷积层有着跳线连接结构，将两部分具有同样尺寸的特征(要输入卷积层的特征和对应的反卷积层输出的特征)做相加操作(ResNet那样的操作)后再输入到下一个反卷积层。这样的结构能够让反向传播信号能够直接传递到底层，解决了梯度消失问题，同时能将卷积层的细节传递给反卷积层，能够恢复出更干净的图片。可以看到，网络中有一条线是将输入的图像连接到后面与最后的一层反卷积层的输出相加，也就是VDSR中用到的方式，因此RED中间的卷积层和反卷积层学习的特征是目标图像和低质图像之间的残差。RED的网络深度为30层，损失函数用的均方误差。

**7. DRRN**

([Image Super-Resolution via Deep Recursive Residual Network](http://openaccess.thecvf.com/content_cvpr_2017/papers/Tai_Image_Super-Resolution_via_CVPR_2017_paper.pdf), CVPR2017)

github(caffe): <https://github.com/tyshiwo/DRRN_CVPR17>

DRRN的作者应该是受到了ResNet、VDSR和DRCN的启发，采用了更深的网络结构来获取性能的提升。作者也在文中用图片示例比较了DRRN与上述三个网络的区别，比较示例图如下所示。



DRRN中的每个残差单元都共同拥有一个相同的输入，即递归块中的第一个卷积层的输出。每个残差单元都包含2个卷积层。在一个递归块内，每个残差单元内对应位置相同的卷积层参数都共享(图中DRRN的浅绿色块或浅红色块)。作者列出了ResNet、VDSR、DRCN和DRRN四者的主要策略。ResNet是链模式的局部残差学习。VDSR是全局残差学习。DRCN是全局残差学习+单权重的递归学习+多目标优化。DRRN是多路径模式的局部残差学习+全局残差学习+多权重的递归学习。

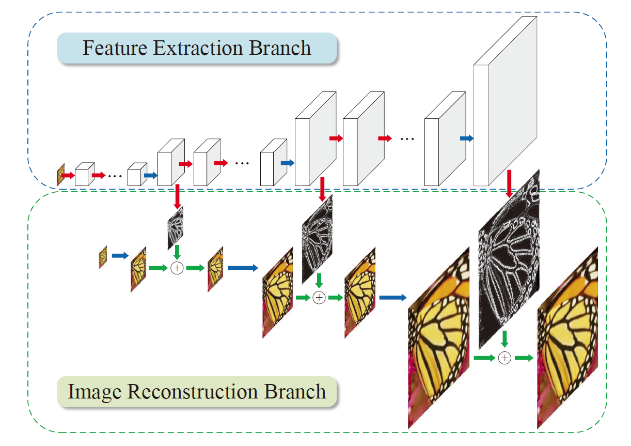
文章中比较了不同的递归块和残差单元数量的实验结果，最终选用的是1个递归块和25个残差单元，深度为52层的网络结构。总之，DRRN就是通过对之前已有的ResNet等结构进行调整，采取更深的网络结构得到结果的提升。

**8. LapSRN**

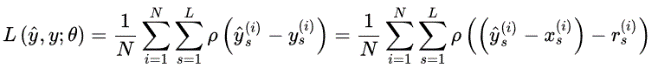
([Deep Laplacian Pyramid Networks for Fast and Accurate Super-Resolution](https://arxiv.org/abs/1704.03915), CVPR2017)

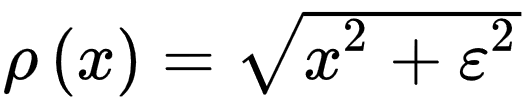
github(tensorflow): <https://github.com/zjuela/LapSRN-tensorflow>

论文中作者先总结了之前的方法存在有三点问题。一是有的方法在输入图像进网络前，需要使用预先定义好的上采样操作(例如bicubic)来获得目标的空间尺寸，这样的操作增加了额外的计算开销，同时也会导致可见的重建伪影。而有的方法使用了亚像素卷积层或者反卷积层这样的操作来替换预先定义好的上采样操作，这些方法的网络结构又相对比较简单，性能较差，并不能学好低分辨率图像到高分辨率图像复杂的映射。二是在训练网络时使用型损失函数时，不可避免地会产生模糊的预测，恢复出的高分辨率图片往往会太过于平滑。三是在重建高分辨率图像时，如果只用一次上采样的操作，在获得大倍数(8倍以上)的上采样因子时就会比较困难。而且在不同的应用时，需要训练不同上采样倍数的模型。针对这三点问题，作者提出了LapSRN，网络结构如下图所示。



LapSRN的结构可以看成有多级，每一级完成一次2倍的上采样操作，要实现8倍的上采样就需要有三级。在每一级中，先通过一些级联的卷积层提取特征，接着通过一个反卷积层将提取出的特征的尺寸上采样2倍。反卷积层后连有两个卷积层，一个卷积层的作用是继续提取特征，另外一个卷积层的作用是预测出这一级的残差。输入图像在每一级也经过一个反卷积层使尺寸上采样2倍，再与对应级的残差相加，就能重构出这一级的上采样结果。LapSRN设计损失函数为：



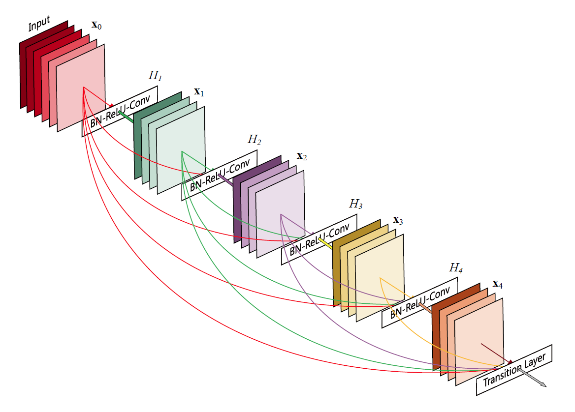
其中, 叫作Charbonnier惩罚函数(范数的变形)，大小设置为0.001。x表示低分辨率图像，y表示高分辨率图像，r表示残差，s表示对应的级。N表示训练时batch size的大小，L表示网络一共有多少级。通过将高分辨率图下采样，在每一级都存在有对应的ground truth进行监督，因此每一级都有一个损失，训练的时候就是要把每一级的损失的和降低。

LapSRN通过逐步上采样，一级一级预测残差的方式，在做高倍上采样时，也能得到中间低倍上采样结果的输出。由于尺寸是逐步放大，不是所有的操作都在大尺寸特征上进行，因此速度比较快。LapSRN设计了损失函数来训练网络，对每一级的结果都进行监督，因此取得了不错的结果。

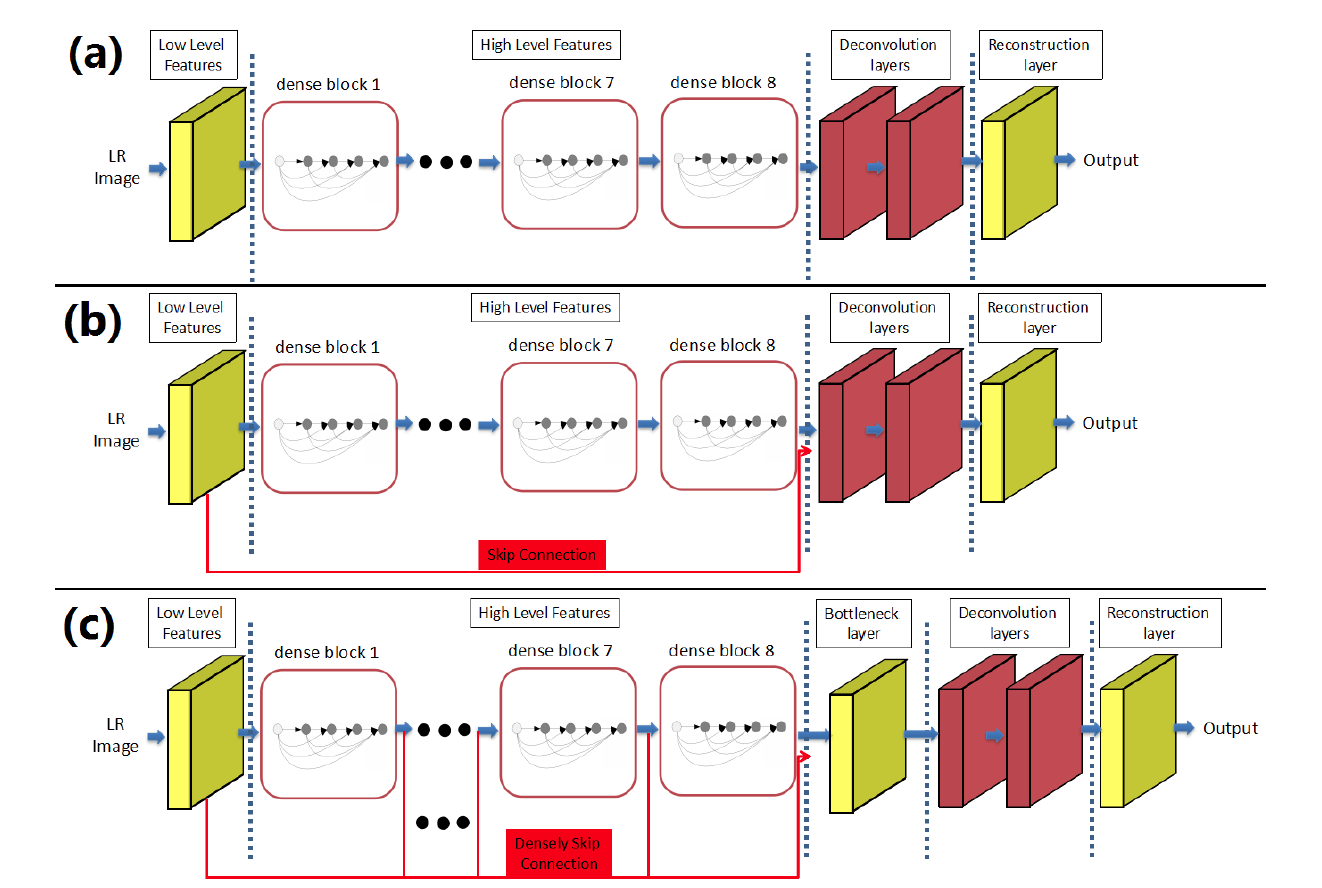
**9. SRDenseNet**

([Image Super-Resolution Using Dense Skip Connections](http://openaccess.thecvf.com/content_ICCV_2017/papers/Tong_Image_Super-Resolution_Using_ICCV_2017_paper.pdf), ICCV2017)

DenseNet是CVPR2017的best papaer获奖论文。DenseNet在稠密块(dense block)中将每一层的特征都输入给之后的所有层，使所有层的特征都串联(concatenate)起来，而不是像ResNet那样直接相加。这样的结构给整个网络带来了减轻梯度消失问题、加强特征传播、支持特征复用、减少参数数量的优点。一个稠密块的结构如下图所示。



SRDenseNet将稠密块结构应用到了超分辨率问题上，取得了不错的效果。网络结构如下图所示。



SRDenseNet可以分成四个部分。首先是用一个卷积层学习低层的特征，接着用多个稠密块学习高层的特征，然后通过几个反卷积层学到上采样滤波器参数，最后通过一个卷积层生成高分辨率输出。

文章中针对用于最后重建的输入内容不同，设计了三种结构并做了比较。一是反卷积层只输入最顶层稠密块的输出。二是添加了一个跳跃连接，将最底层卷积层的输出特征和最顶层稠密块的输出特征串联起来，再输入反卷积层。三是添加了稠密跳跃连接，就是把稠密块看成一个整体，第一个卷积层的输出以及每个稠密块的输出，都输入给在之后的所有稠密块，像是把在反卷积层之前的整个网络也设计成像稠密块那样的结构。由于这样做，所有的特征都串联起来，这样直接输入反卷积层会产生巨大的计算开销，因此添加了一个核大小为1×1的卷积层来减小特征数量，这个卷积层被称为瓶颈层。最后的结果是越复杂的越好，3>2>1。文章中分析的是，受益于低层特征和高层特征的结合，超分辨率重建的性能得到了提升。像第三种结构把所有深度层的特征都串联起来，得到了最佳的结果，说明不同深度层的特征之间包含的信息是互补的。

**10. SRGAN(SRResNet)**

([Photo-Realistic Single Image Super-Resolution Using a Generative Adversarial Network](https://arxiv.org/abs/1609.04802), CVPR2017)

github(tensorflow): <https://github.com/zsdonghao/SRGAN>

github(tensorflow): <https://github.com/buriburisuri/SRGAN>

github(tensorflow): <https://github.com/brade31919/SRGAN-tensorflow>[https://RGAN-tensorflow](http://link.zhihu.com/?target=https%3A//github.com/brade31919/SRGAN-tensorflow)

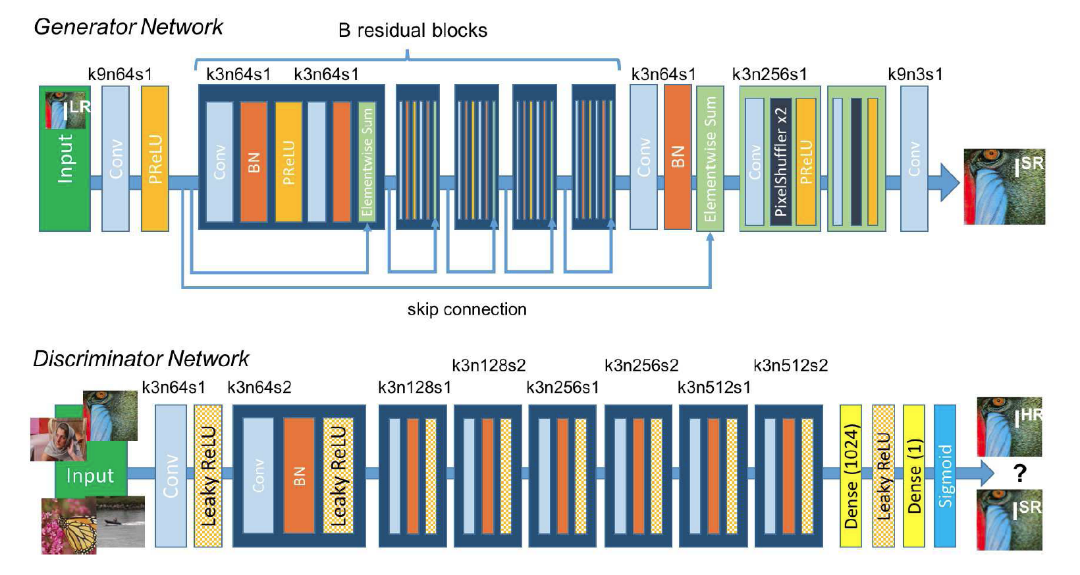
github(keras): <https://github.com/titu1994/Super-Resolution-using-Generative-Adversarial-Networks>[https://er-Resolution-using-Generative-Adversarial-Networks](http://link.zhihu.com/?target=https%3A//github.com/titu1994/Super-Resolution-using-Generative-Adversarial-Networks)

在这篇文章中，将生成对抗网络(Generative Adversarial Network, GAN)用在了解决超分辨率问题上。文章提到，训练网络时用均方差作为损失函数，虽然能够获得很高的峰值信噪比，但是恢复出来的图像通常会**丢失高频细节**，使人不能有好的视觉感受。SRGAN利用感知损失(perceptual loss)和对抗损失(adversarial loss)来提升恢复出的图片的真实感。感知损失是利用卷积神经网络提取出的特征，通过比较生成图片经过卷积神经网络后的特征和目标图片经过卷积神经网络后的特征的差别，使生成图片和目标图片在语义和风格上更相似。

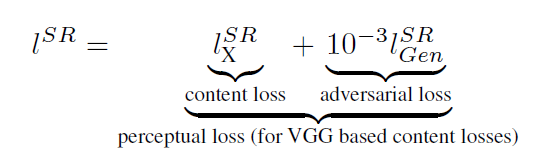
一个GAN所要完成的工作，GAN原文举了个例子：生成网络(G)是印假钞的人，判别网络(D)是检测假钞的人。G的工作是让自己印出来的假钞尽量能骗过D，D则要尽可能的分辨自己拿到的钞票是银行中的真票票还是G印出来的假票票。开始的时候呢，G技术不过关，D能指出这个假钞哪里很假。G每次失败之后都认真总结经验，努力提升自己，每次都进步。直到最后，D无法判断钞票的真假……

SRGAN的工作就是： G网通过低分辨率的图像生成高分辨率图像，由D网判断拿到的图像是由G网生成的，还是数据库中的原图像。当G网能成功骗过D网的时候，那我们就可以通过这个GAN完成超分辨率了。

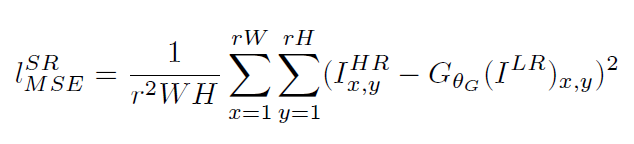
文章中，用**均方误差优化SRResNet**(SRGAN的生成网络部分)，能够得到具有很高的峰值信噪比的结果。在训练好的VGG模型的高层特征上**计算感知损失**来优化SRGAN，并结合SRGAN的判别网络，能够得到峰值信噪比虽然不是最高，但是具有逼真视觉效果的结果。SRGAN网络结构如下图所示。



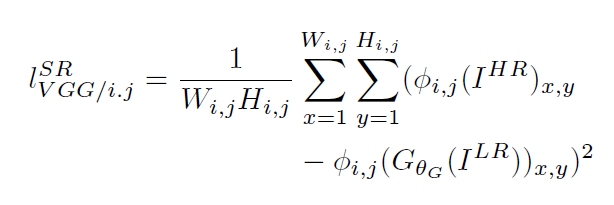
在**生成网络部分**(SRResNet)包含多个残差块，每个残差块中包含两个3×3的卷积层，卷积层后接批规范化层(batch normalization, BN)和PReLU作为激活函数，两个2×亚像素卷积层(sub-pixel convolution layers)被用来增大特征尺寸。在判别网络部分包含8个卷积层，随着网络层数加深，特征个数不断增加，**特征尺寸不断减小**，选取激活函数为LeakyReLU，最终通过两个全连接层和最终的sigmoid激活函数得到**预测为自然图像的概率**。SRGAN的损失函数为：



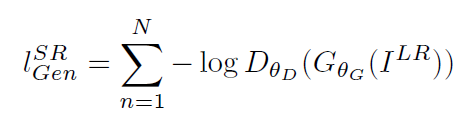
其中内容损失可以是基于均方误差的损失函数：



也可以是基于训练好的以ReLU为激活函数的VGG模型的损失函数:



i和j表示VGG19网络中第i个最大池化层(maxpooling)后的第j个卷积层得到的特征。对抗损失为：



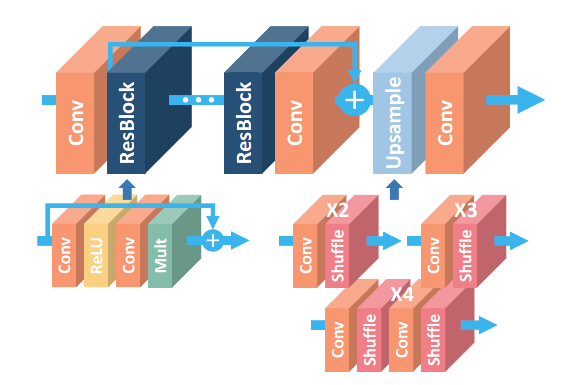
文章中的实验结果表明，用基于均方误差的损失函数训练的SRResNet，得到了结果具有很高的峰值信噪比，但是会**丢失一些高频部分细节**，图像比较平滑。而SRGAN得到的结果则有更好的视觉效果。其中，又对内容损失分别设置成基于均方误差、基于VGG模型低层特征和基于VGG模型高层特征三种情况作了比较，在基于均方误差的时候表现最差，**基于VGG模型高层特征比基于VGG模型低层特征的内容损失能生成更好的纹理细节**。

**11. EDSR**

([Enhanced Deep Residual Networks for Single Image Super-Resolution](https://arxiv.org/abs/1707.02921), CVPR2017)

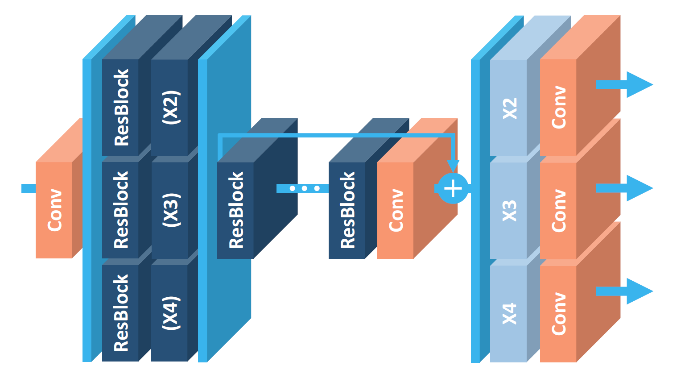
github(tensorflow): <https://github.com/jmiller656/EDSR-Tensorflow>

EDSR是NTIRE2017超分辨率挑战赛上获得冠军的方案。如论文中所说，EDSR最有意义的模型性能提升是去除掉了SRResNet多余的模块，从而可以**扩大模型的尺寸**来提升结果质量。EDSR的网络结构如下图所示。



可以看到，EDSR在结构上与SRResNet相比，就是把批规范化处理(batch normalization, BN)操作给去掉了。文章中说，原始的ResNet最一开始是被提出来解决高层的计算机视觉问题，比如分类和检测，直接把ResNet的结构应用到像超分辨率这样的低层计算机视觉问题，显然不是最优的。由于批规范化层消耗了与它前面的卷积层相同大小的内存，在去掉这一步操作后，相同的计算资源下，EDSR就可以堆叠更多的网络层或者使每层提取更多的特征，从而得到更好的性能表现。EDSR用L1范数样式的损失函数来优化网络模型。在训练时先训练低倍数的上采样模型，接着用训练低倍数上采样模型得到的参数来初始化高倍数的上采样模型，这样能减少高倍数上采样模型的训练时间，同时训练结果也更好。

这篇文章还提出了一个能同时不同上采样倍数的网络结构MDSR，如下图。



MDSR的中间部分还是和EDSR一样，只是在网络前面添加了不同的预训练好的模型来减少不同倍数的输入图片的差异。在网络最后，不同倍数上采样的结构平行排列来获得不同倍数的输出结果。

从文章给出的结果可以看到，EDSR能够得到很好的结果。增大模型参数数量以后，结果又有了进一步的提升。因此如果能够解决训练困难的问题，网络越深，参数越多，对提升结果确实是有帮助吧。