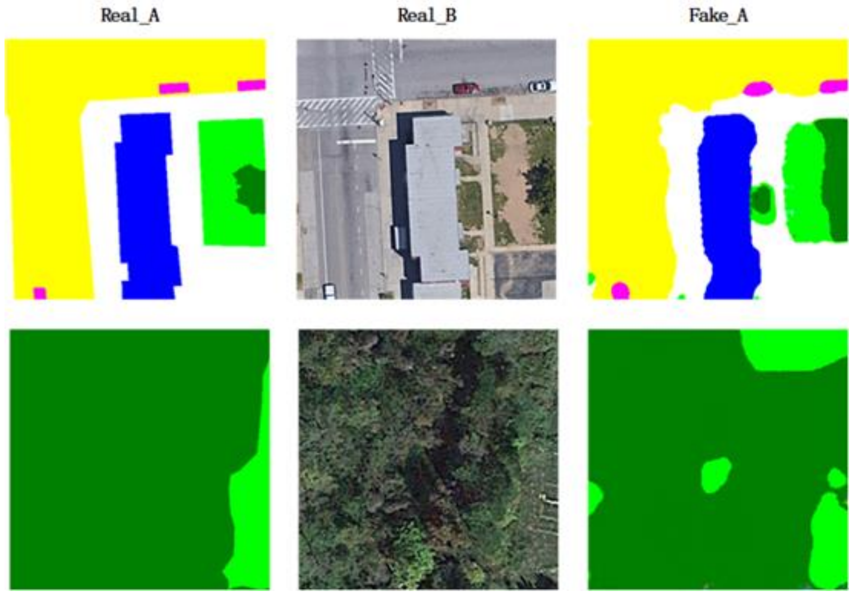
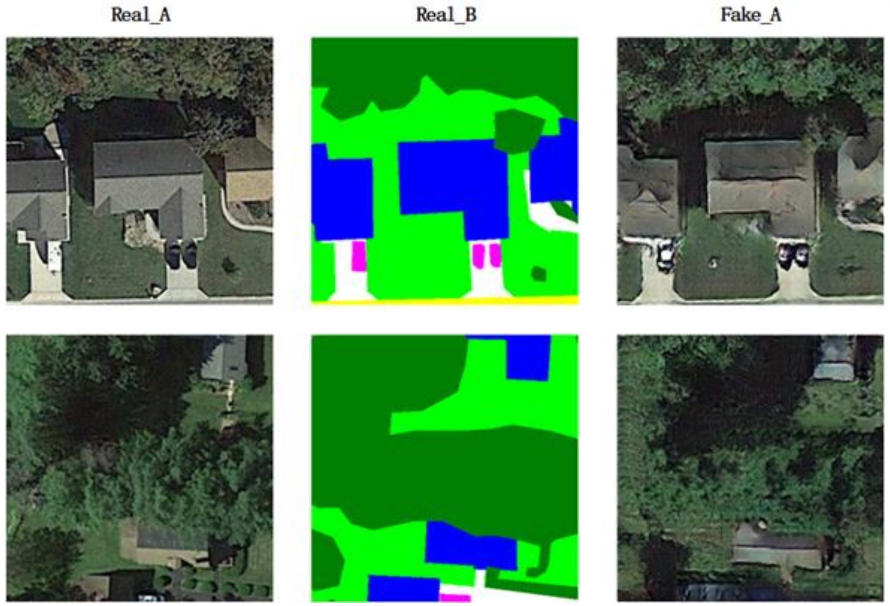
Image-to-Image Translation

PIX2PIX





演示人：李嘉鑫

学院：宇航学院

日期：2020年11月20日

数据处理

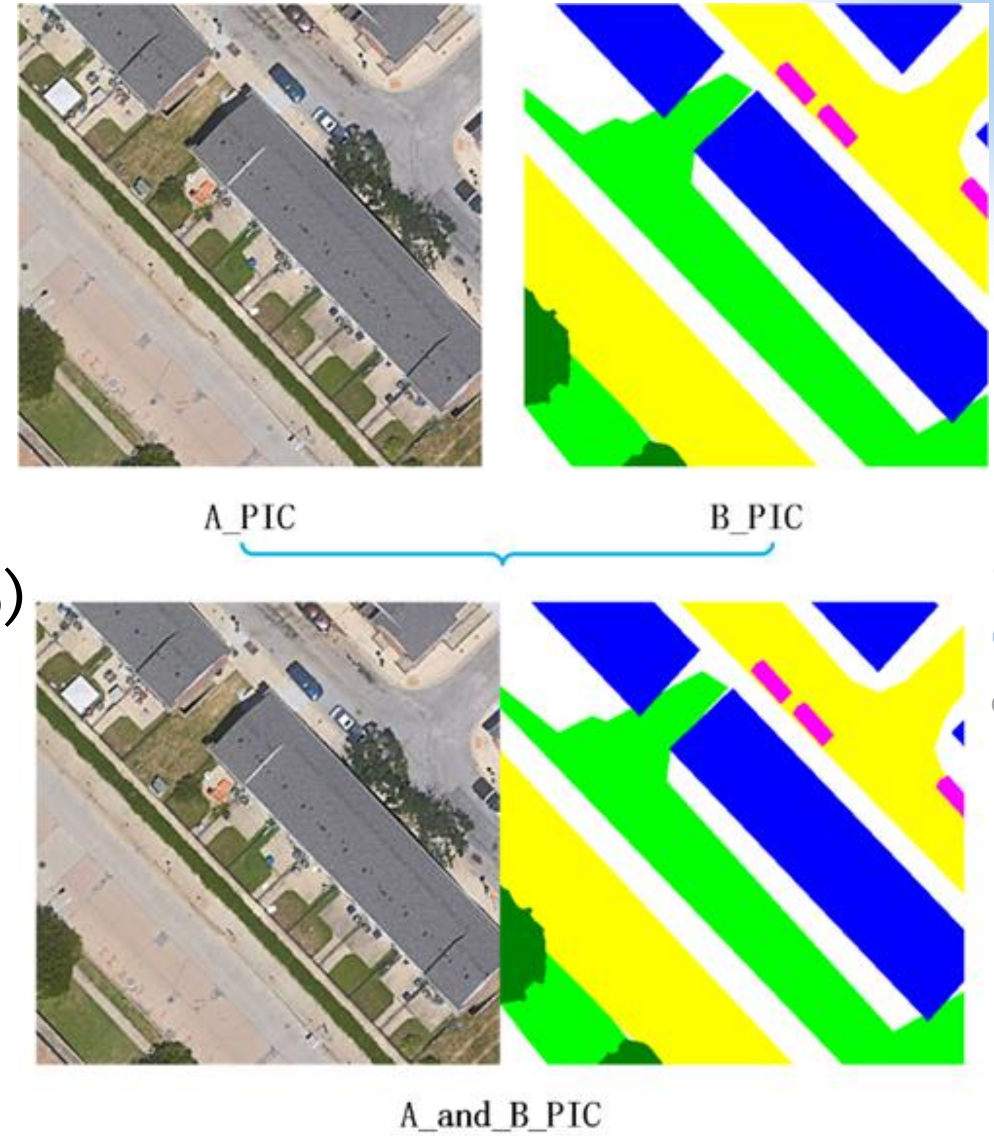
1. A图与B图进行合并

2. 划分数据集——8:1:1

3. 训练中，尺寸调整 Resize(286,286)

4. 随机裁剪 Randomcrop(256,256)

5. 归一化 [-1,1]



Pix2Pix-生成器

(256,256,3)				(256,256,3)
Conv(3,64,4,2,1)		uprelu, upConv(128,3,4,2,1),nn.Tanh		
(128,128,64)				(128,128,64)
downrelu,Conv(64,128,4,2,1),downnorm		uprelu,upConv(256,64,4,2,1),upnorm		
(64,64,128)	concat	(64,64,128)		(64,64,256)
downrelu,Conv(128,256,4,2,1),downnorm		uprelu,upConv(512,128,4,2,1),upnorm		
(32,32,256)	concat	(32,32,256)		(32,32,512)
downrelu,Conv(256,512,4,2,1),downnorm		uprelu,upConv(1024,256,4,2,1),upnorm		
(16,16,512)	concat	(16,16,512)		(16,16,1024)
downrelu,Conv(512,512,4,2,1),downnorm		uprelu,upConv(1024,512,4,2,1),upnorm		
(8,8,512)	concat	(8,8,512)		(8,8,1024)
downrelu,Conv(512,512,4,2,1),downnorm		uprelu,upConv(1024,512,4,2,1),upnorm		
(4,4,512)	concat	(4,4,512)		(4,4,1024)
downrelu,Conv(512,512,4,2,1),downnorm		uprelu,upConv(1024,512,4,2,1),upnorm		
(2,2,512)	concat	(2,2,512)		(2,2,1024)
downrelu,Conv(512,512,4,2,1)		uprelu,upConv(512,512,4,2,1),upnorm		
(1,1,512)				

Downrelu => nn.LeakyReLU(0.2)

Norm => nn.InstanceNorm

Pix2Pix-判别器

(256,256,6)
Conv(6,64,4,2,1), LeakyReLU(0.2)
(128,128,64)
Conv(64,128,4,2,1),InstanceNorm, LeakyReLU(0.2)
(64,64,128)
Conv(128,256,4,2,1),InstanceNorm, LeakyReLU(0.2)
(32,32,256)
Conv(256,512,4,1,1),InstanceNorm, LeakyReLU(0.2)
(32,32,512)
Conv(512,1,4,1,1),InstanceNorm, LeakyReLU(0.2)
(32,32,1)

判别器采用patchGAN中的判别器，因为最后特征图对应的感受野是70x70的小块

损失函数及优化器

损失函数：

1）GANloss：采用二值交叉熵

2）L1loss：采用L1损失，相比L2损失在接近零的时候变化较为明显

1）生成器：

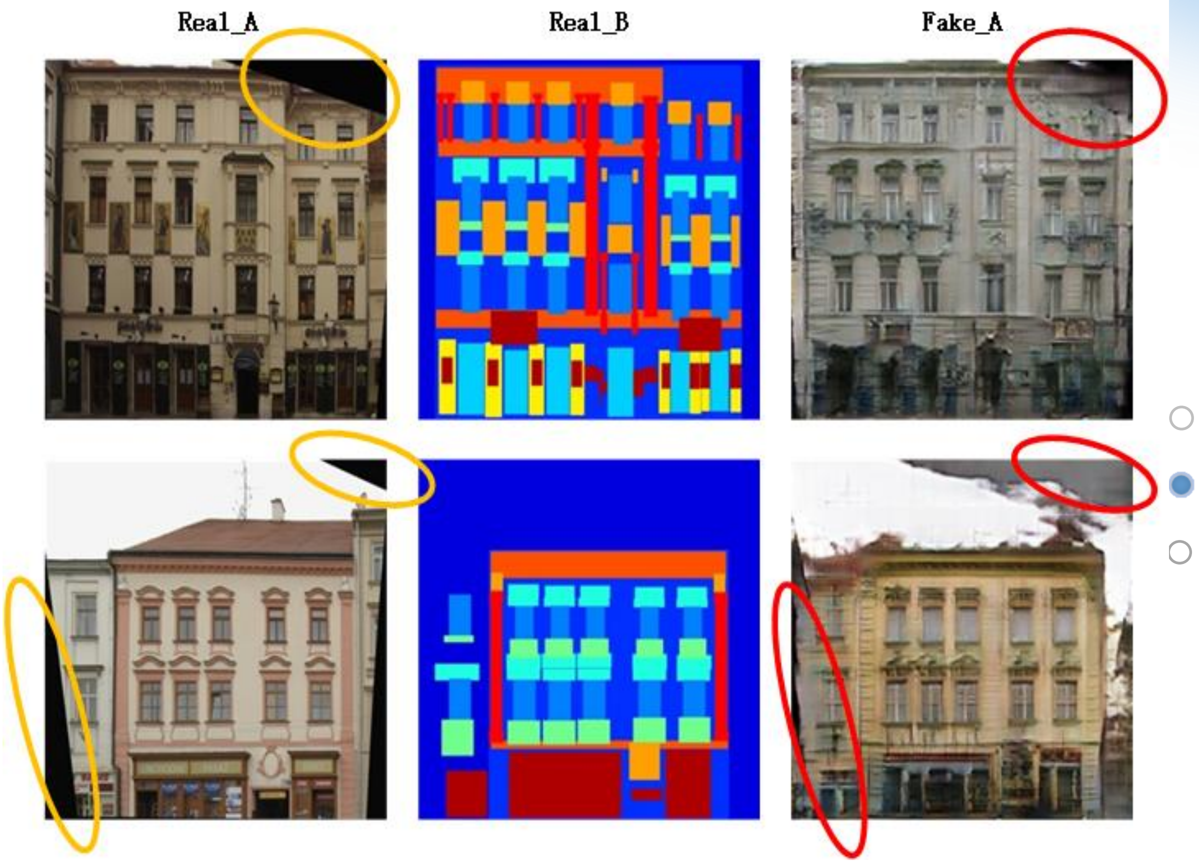
loss_G=(GANloss(real_A, fake_B)+L1loss(fake_B, real_B))*lmadaL1

2）判别器：

loss_D=(GANloss(real_A, fake_B)+GANloss(real_A,real_B))*0.5

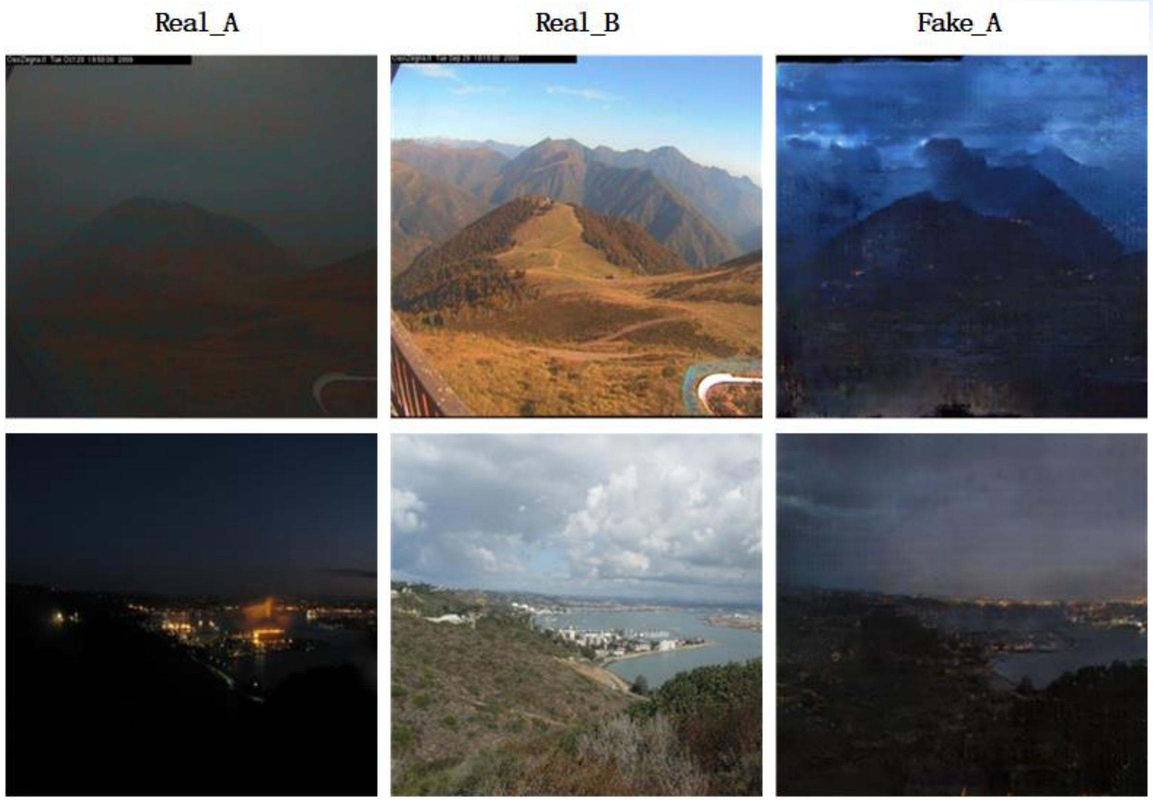
Facades数据集测试

通过观察我们可以发现，
四个角落的生成效果较差



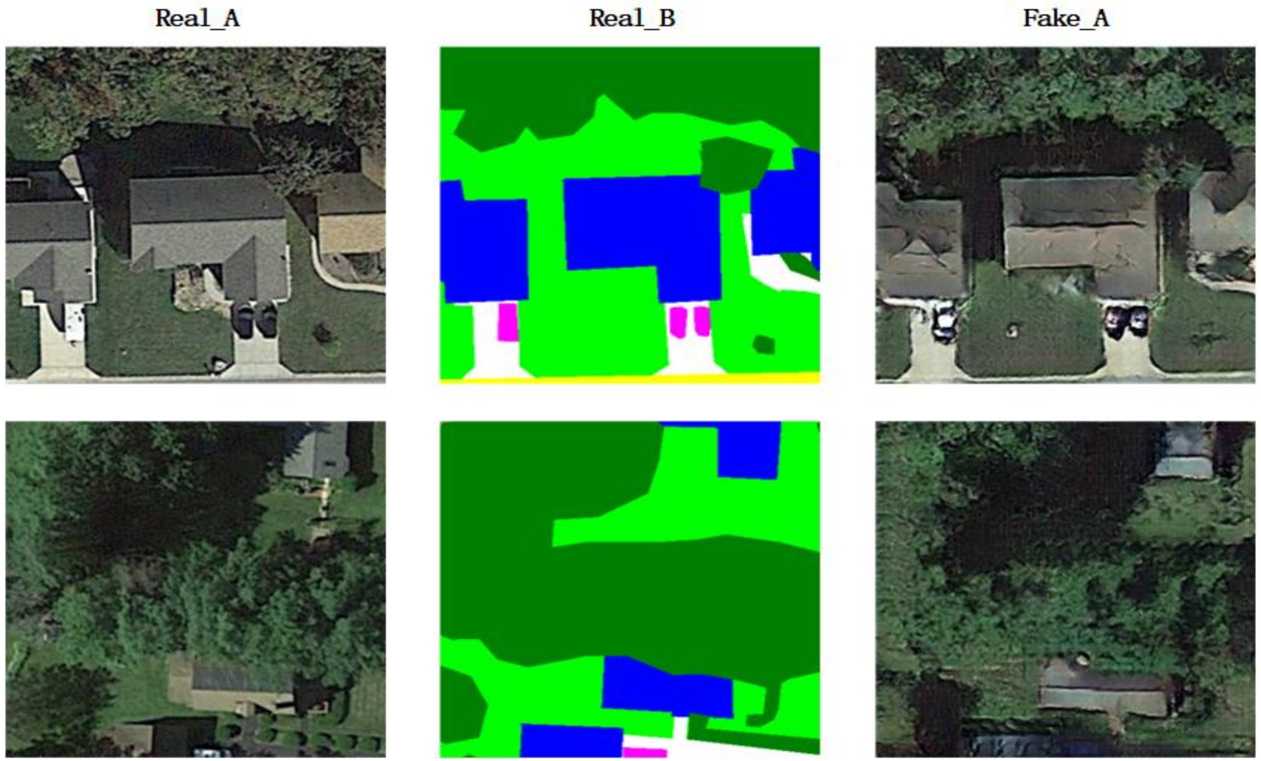
day2night数据集测试

网络图片



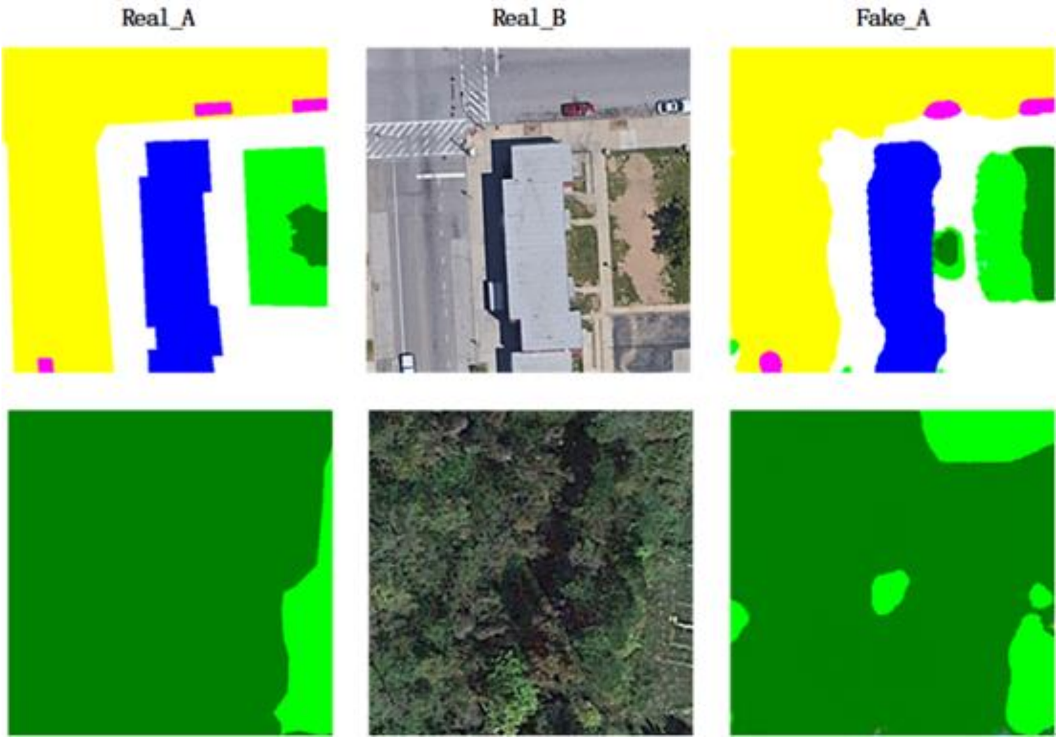
Day2night测试集图片

中科星图语义分割数据集训练

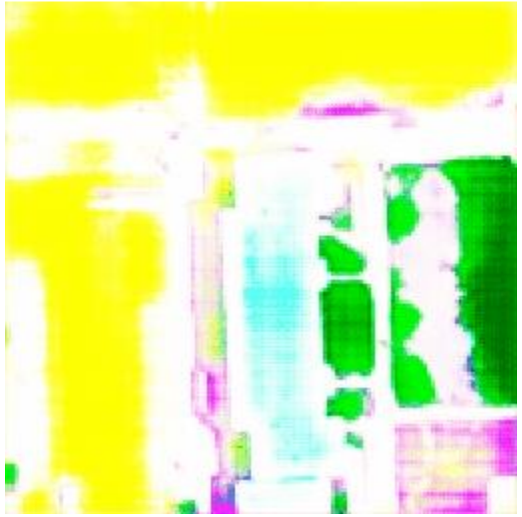


语义地图转换真实地图

中科星图语义分割数据集训练



真实地图转换语义地图



5-200epoch的测试图

语义分割综述——讲一半

FCNs

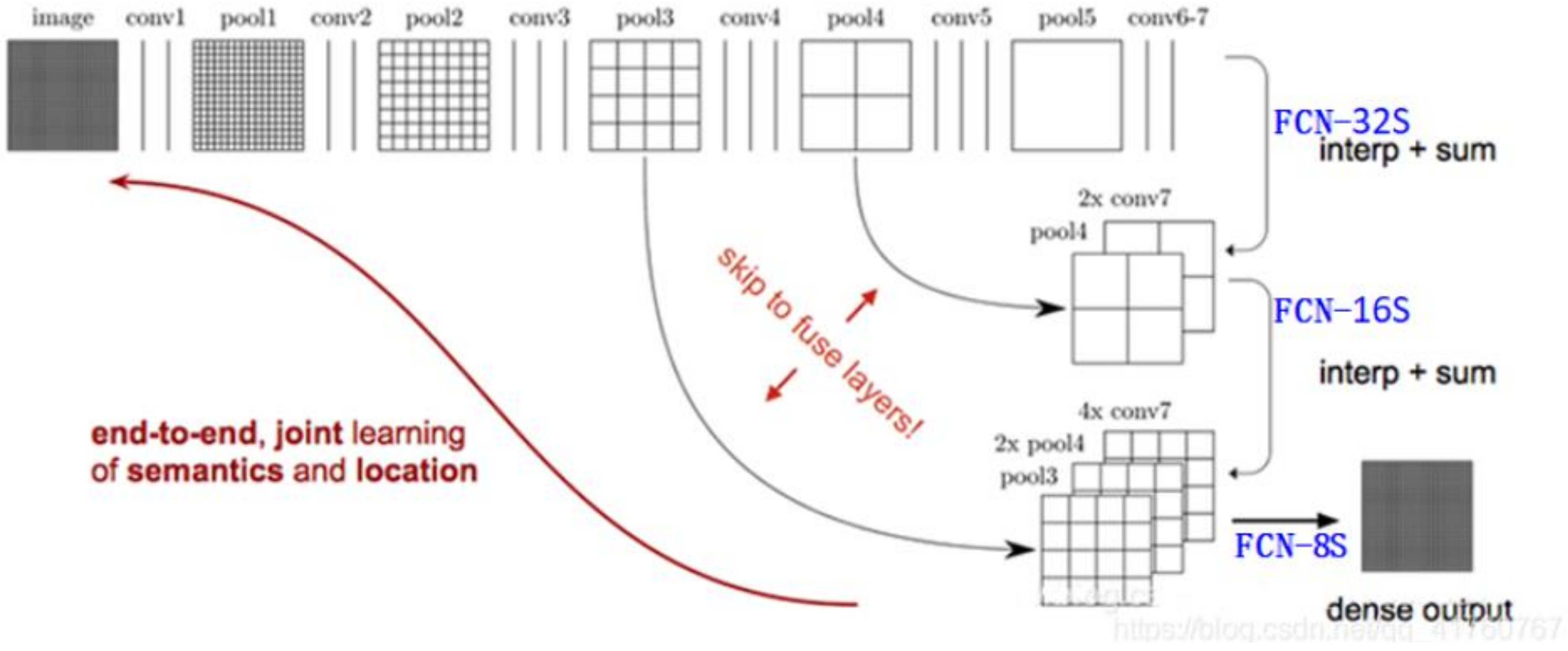
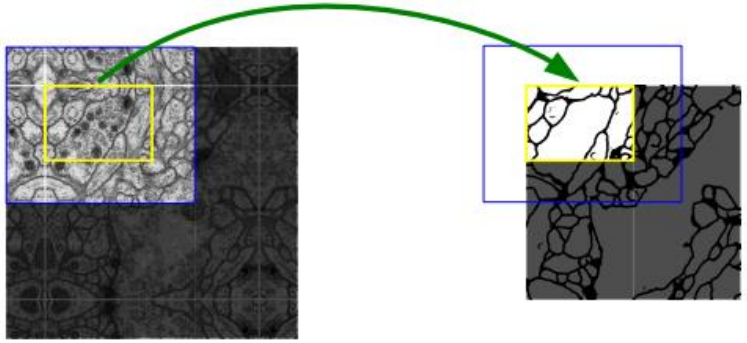
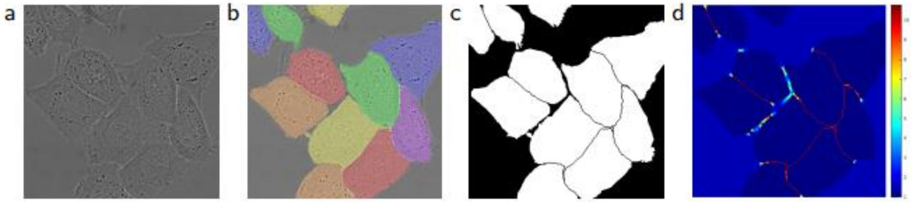


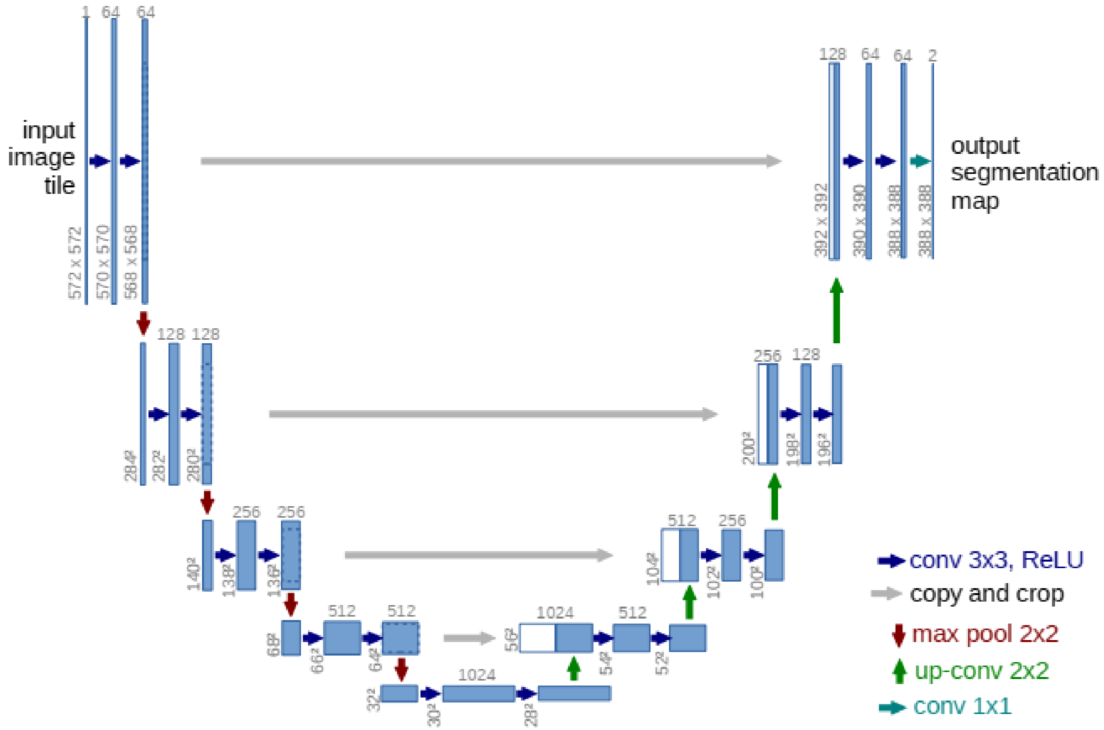
image
conv1
pool1
conv2
pool2
conv3
pool3
conv4
pool4
conv5
pool5
conv6-7
FCN-32S
interp + sum
2x conv7
pool4
FCN-16S
interp + sum
4x conv7
2x pool4
pool3
FCN-8S
dense output
skip to fuse layers!
end-to-end, joint learning
of semantics and location

U-net





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



SegNet

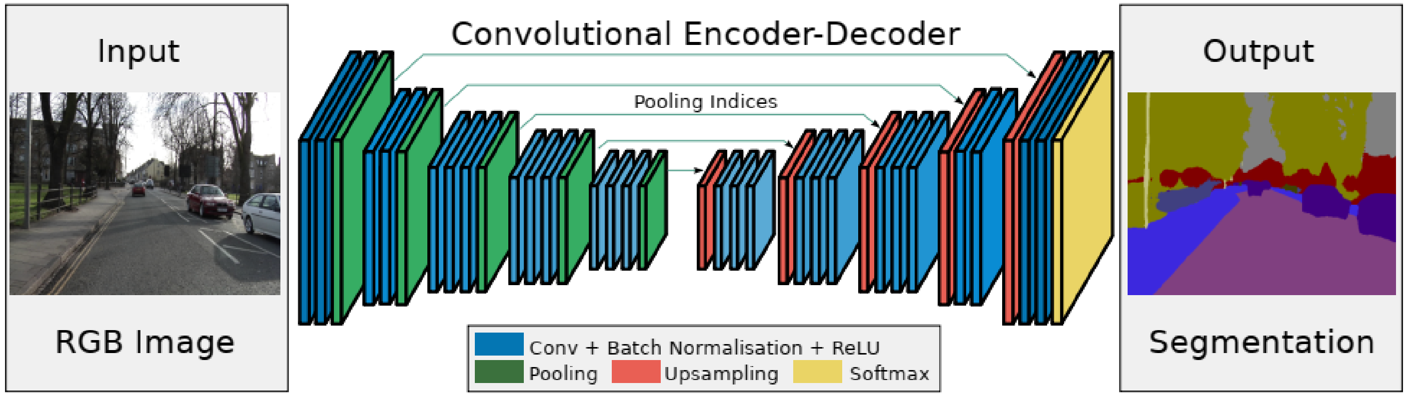
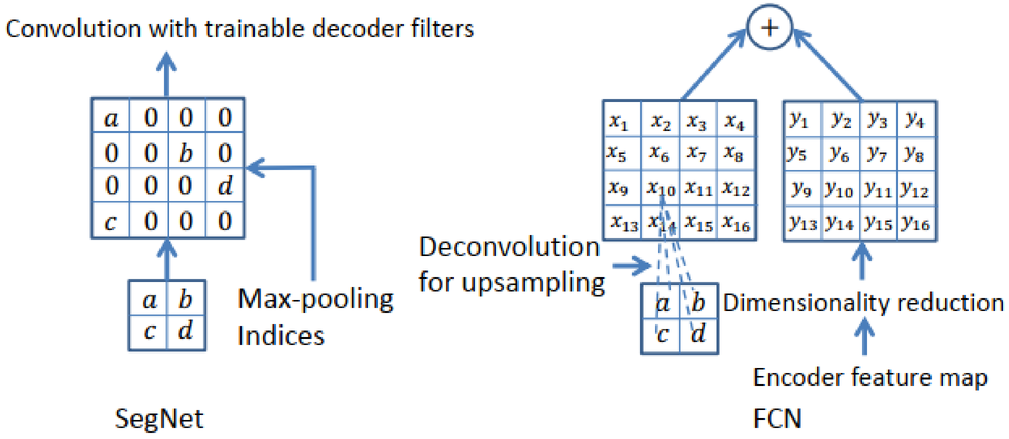
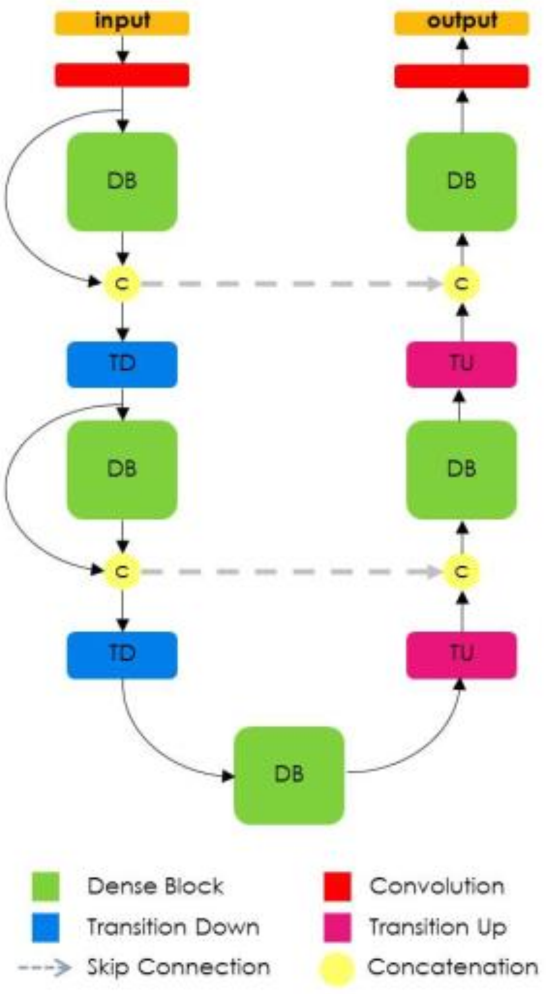


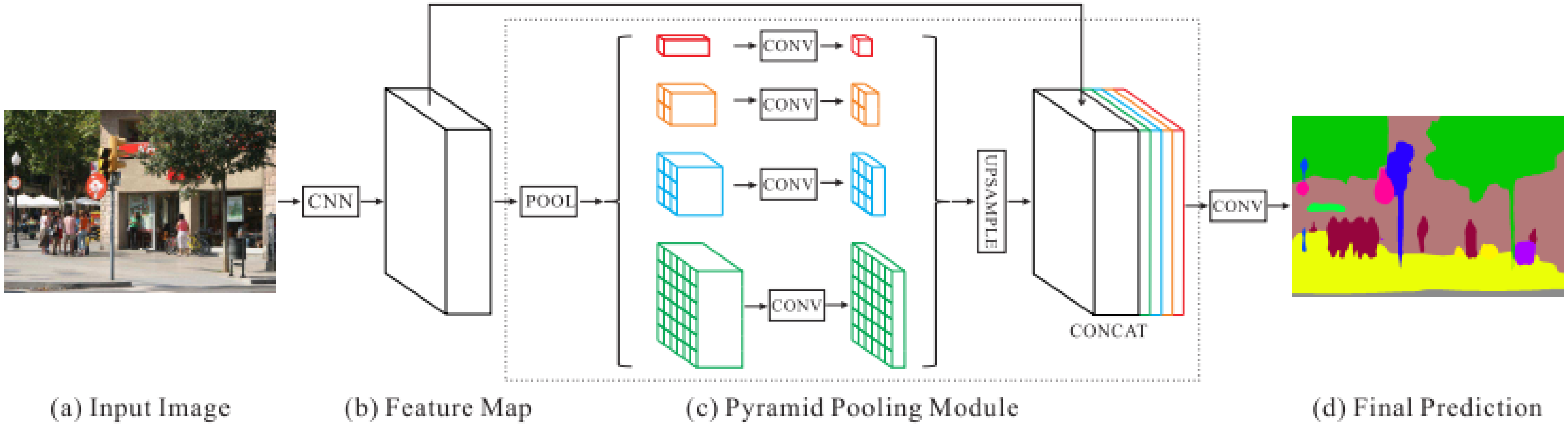
Fig. 2. An illustration of the SegNet architecture. There are no fully connected layers and hence it is only convolutional. A decoder upsamples its input using the transferred pool indices from its encoder to produce a sparse feature map(s). It then performs convolution with a trainable filter bank to densify the feature map. The final decoder output feature maps are fed to a soft-max classifier for pixel-wise classification.



Fully Convolutional DenseNet



PSPNet



CNN
POOL
CONV
CONV
CONV
CONV
UPSAMPLE
CONCAT
CONV
(a) Input Image
(b) Feature Map
(c) Pyramid Pooling Module
(d) Final Prediction

谢谢大家聆听

联系方式：
邮箱：vpmaster1@163.com
微信：ber2016ber