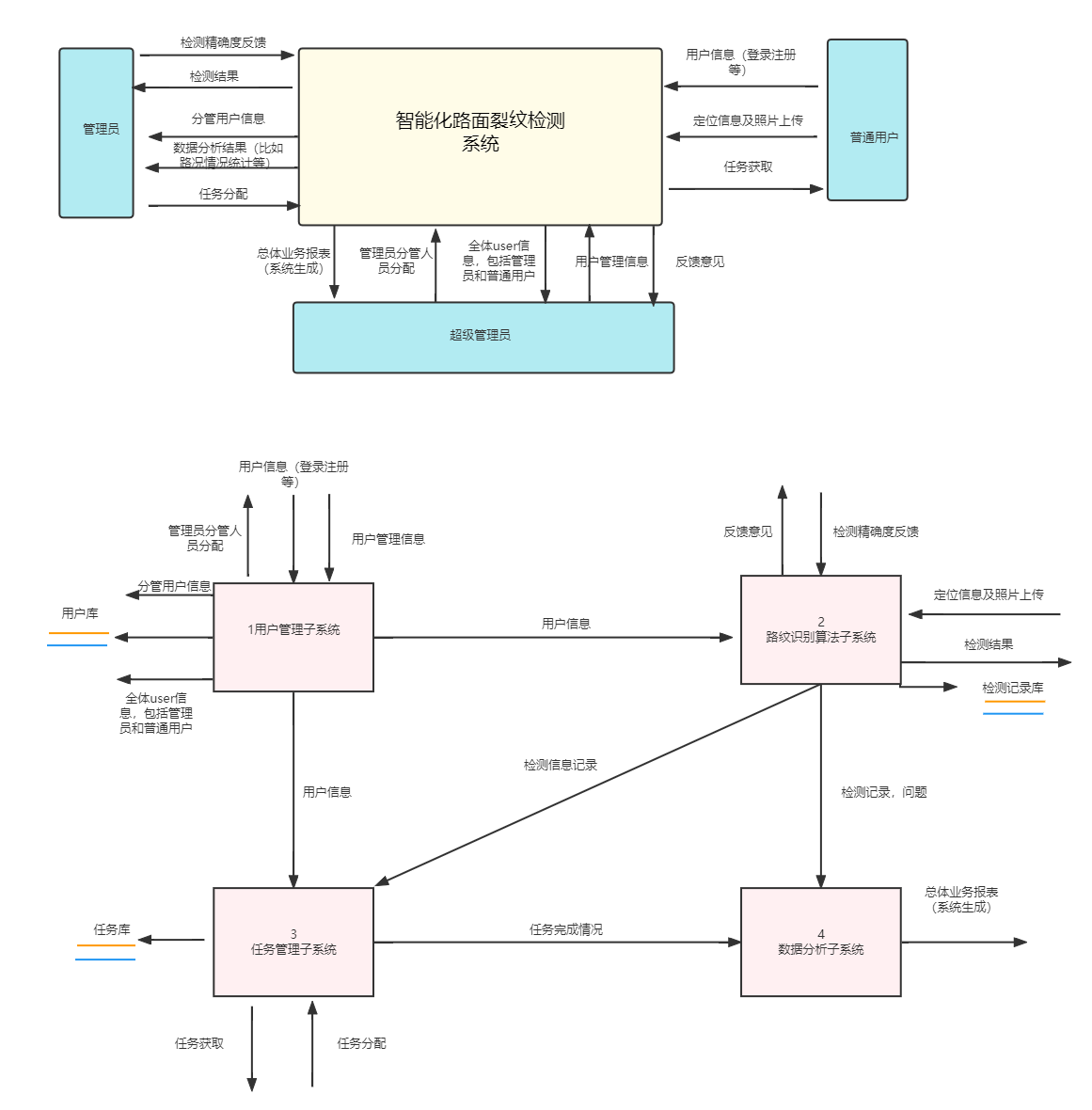
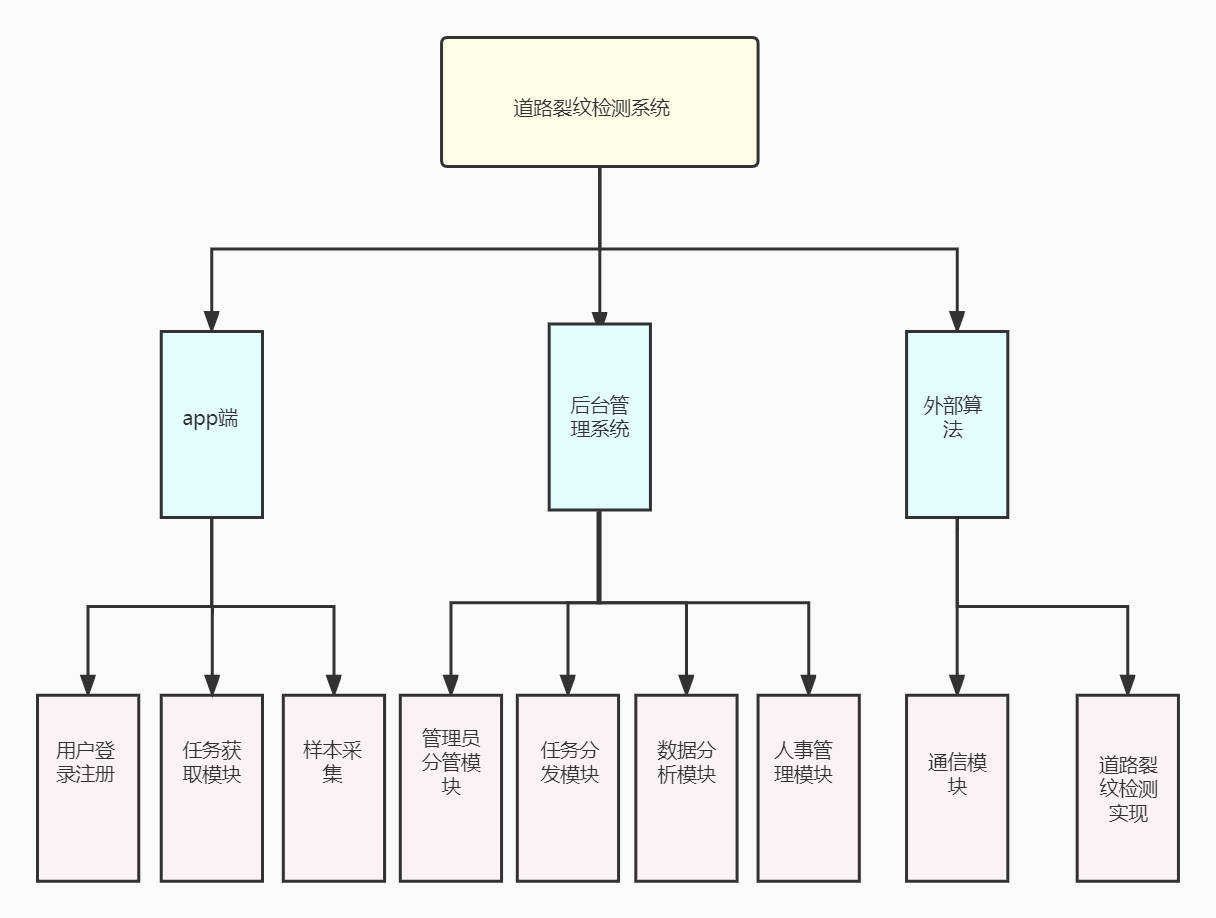
**智能化路面裂缝检测系统功能**

1. **系统框架图**
2. **数据流图**

****

**2 功能模块图**



智能化路面裂纹检测系统

**3 状态图**



登陆成功

执行任务

登陆APP 获取任务 上传照片

执行算法

登陆失败

完成任务

登陆成功

登陆后台

任务分配

获取定位

登陆成功

完成任务后

发送信息

人事管理

反馈信息

记录任务

分配管理员

**二．功能说明**

**1.APP模块**

1.1 开发环境：JAVAWEB，系统能够适配ios和安卓不同的平台。

1.2 用户管理子模块

普通用户注册登录app端接受任务并执行任务。普通用户无权发布任务。

1.3 接收任务子模块

用户接收到由管理员分发的任务，并且可以查看自己完成的和没有完成的任务。

1.4 采集子模块

用户接到任务之后，可以去到任务地点进行拍摄工作。拍摄时可以自动获取位置信息，包括经纬度等，位置信息会唯一确定一个地点。







上传照片之后，图片会被发送到算法端，进行裂纹的识别与检测，裂纹识别后被标识出来，然后返回结果。



1.5 复核子模块

管理员和用户对图片进行人工复核，查看裂纹检测的结果是否精准，是否存在漏检、误检的情况，并及时进行反馈以进行算法的进一步优化。

1. **检测模型模块**

2.1 模型功能

针对采集的路面图片进行裂纹检测，将裂纹部分用白色像素标出，而正常路面使用黑色标出。

2.2 运行环境

使用Python3.8，使用torch进行网络搭建。在训练过程中使用CUDA平台在GPU上进行训练，所需显存至少2G以上（即batch为1）。

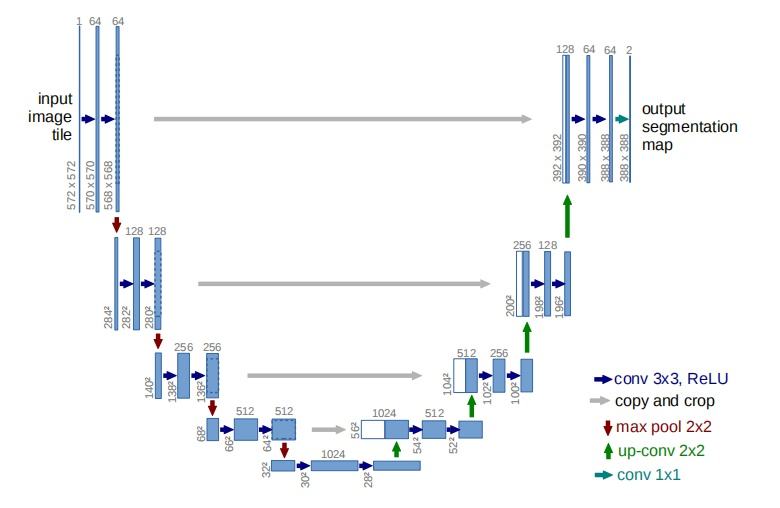
2.3 模型结构

项目选用的模型为U-net，并对其中部分结构进行了改动。Unet 的初衷是为了解决医学[图像分割](https://so.csdn.net/so/search?q=%E5%9B%BE%E5%83%8F%E5%88%86%E5%89%B2&spm=1001.2101.3001.7020)的问题。在解决细胞层面的分割的任务方面，其在 2015 年的 ISBI cell tracking 比赛中获得了多个第一。之后UNet 凭借其突出的分割效果而被广泛应用在语义分割的各个方向。其属于 FCN 的一种变体，简单高效且容易构建。Unet模型优点如下：

（1）深/浅层特征有着各自意义：网络越深，感受野越大，深层网络关注全局特征（更抽象、更本质）；浅层网络则更加关注纹理等局部特征；

（2）通过特征拼接来实现边缘特征的找回。通过上采样（转置卷积）固然能够得到更大尺寸的特征图，但特征图的边缘是缺少信息的。毕竟每一次下采样提取特征的同时，必然会损失一些边缘特征，而上采样并不能找回这些失去的特征。

模型结构如下图所示：



考虑到使用max pool方法会使得前面所采集的特征产生丢失，本项目对于上图中的下采样不再采用max pool而使用卷积的方法。更改后的下采样代码如下：

class DownSample(Module):

def \_\_init\_\_(self, channel):

super(DownSample, self).\_\_init\_\_()

self.layer = nn.Sequential(

nn.Conv2d(channel, channel, 3, 2, 1, padding\_mode='reflect', bias=False), # 这里使用步长为2的3\*3卷积代替池化进行下采样

nn.BatchNorm2d(channel),

nn.LeakyReLU()

)

def forward(self, x):

print(self.layer(x).shape)

return self.layer(x)

2.4 数据处理

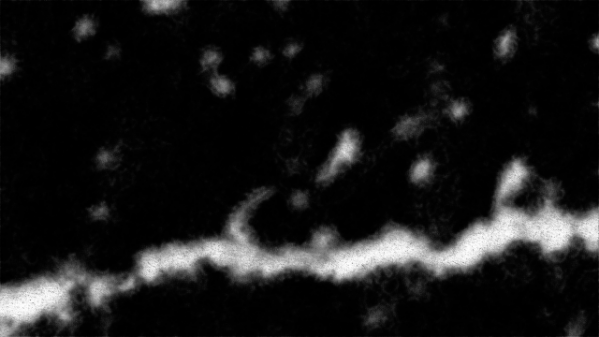
在输入数据的处理方面，首先对图片进行查看。得知大部分图片都是640x360或360x640，所以在这两种分辨率中选择360x640作为输入大小。对于640x360分辨率的图像则对其进行旋转，对于不属于这两类分辨率的图片则对图片调用resize()函数变为360x640的图片。

2.5 训练

可以根据显存大小调节输入的batch，初次训练每50个epcoh保存一次训练参数到一个.pth文件。再次训练时，程序会自动调用之前保存的.pth文件中的训练参数，并在此基础上继续进行训练。

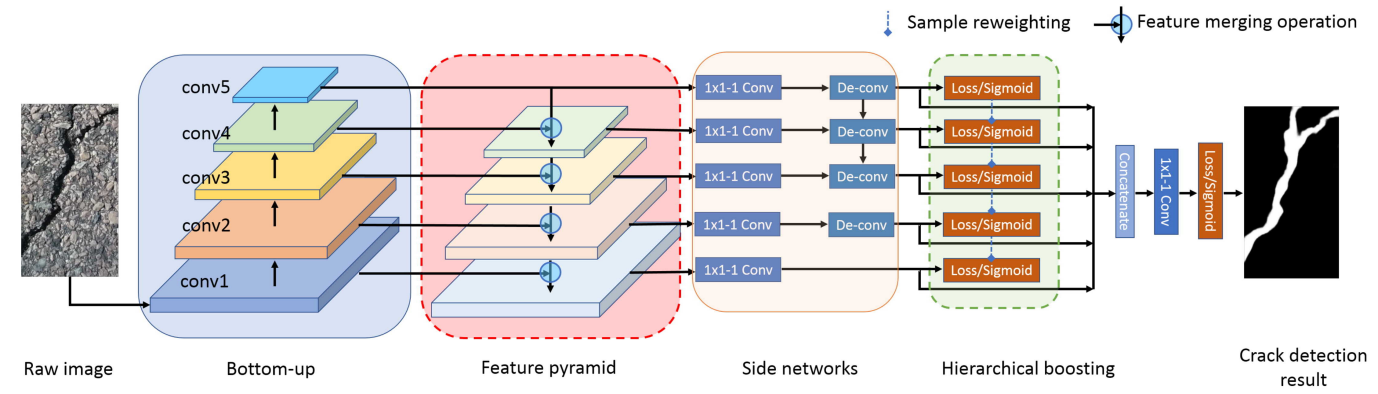
2.6 预测

预测时也同样由代码自动调出.pth中的参数，操作者只需直接将需要预测的图片所在路径输入，便可得到裂纹的预测图片。其原图以及初步训练预测结果如下所示：



2.7对比FPHB

（1）FPHB网络结构：



FPHB以HED为骨干网络，并使用FP（Feature pyramid）将上层特征向下层特征进行传递（通过叠加来完成），并使用HB(Hierarchical boosting)来对训练困难的样本施以更大的权重。

FPHB是2019年提出的，FPHB在论文中认为裂纹检测相比于语义分割问题来说，更像边缘检测问题。FPHB虽然比U-net提出的晚3年左右，但是它在和其他模型做对比的时候并没有把U-net作为对比模型。

虽然FPHB没有提到U-net,但是可以看出就所使用结构上来说，U-net和FPHB都使用了FPN的思想。

1. 二者的对比结果

这里我们使用的FPHB将HB该为使用AdaBoosting,这两者所起的作用都是加大训练困难样本的权重。

对比代码如下：

def count():

model=getModel()

my\_confmat = ConfusionMatrix(2)

compare\_confmat = ConfusionMatrix(2)

loader=getData(batch\_size=8)

for idx,(img,target,compare\_out) in enumerate(loader):

print(f'{idx}/{len(loader)}')

img,target,compare\_out=img.to('cuda'),target.to('cuda'),compare\_out.to('cuda')

out = model(img)['out'].argmax(1)

my\_confmat.update(target.flatten(),out.flatten())

compare\_confmat.update(target.flatten(),compare\_out.flatten())

print('我的网络性能:')

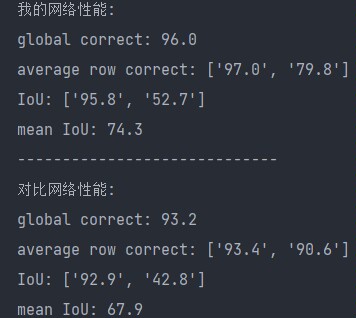
print(my\_confmat)

print('-----------------------------')

print('对比网络性能:')

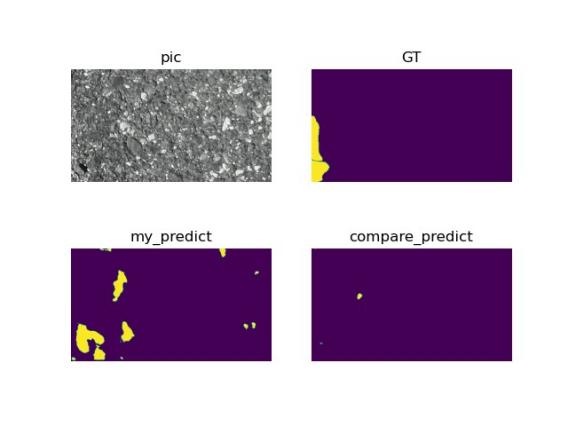
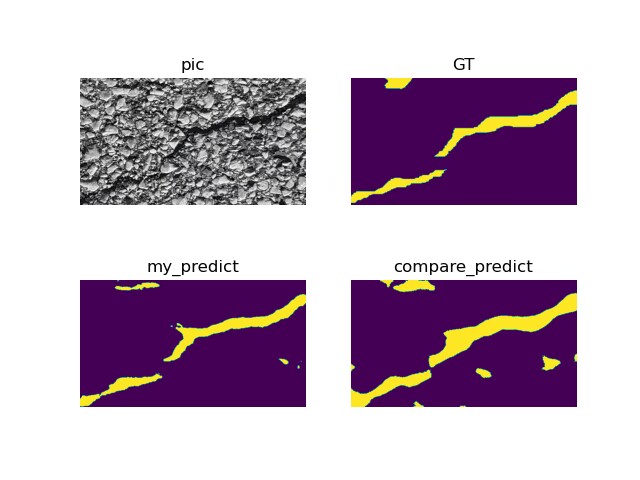
print(compare\_confmat)

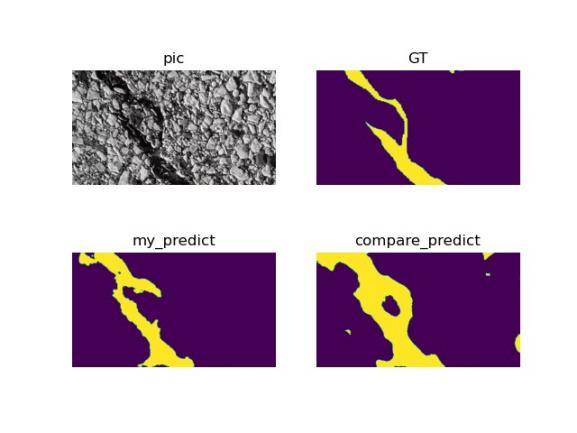
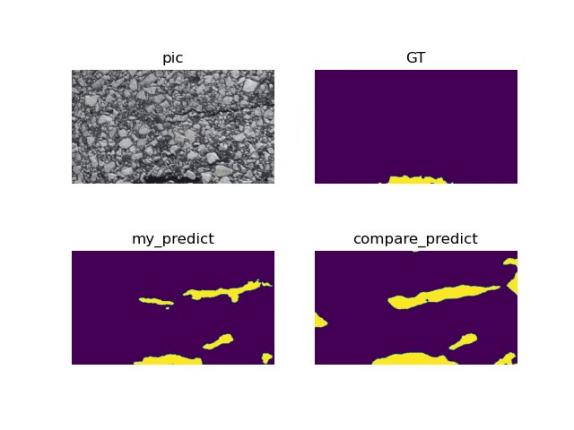
结果如下：



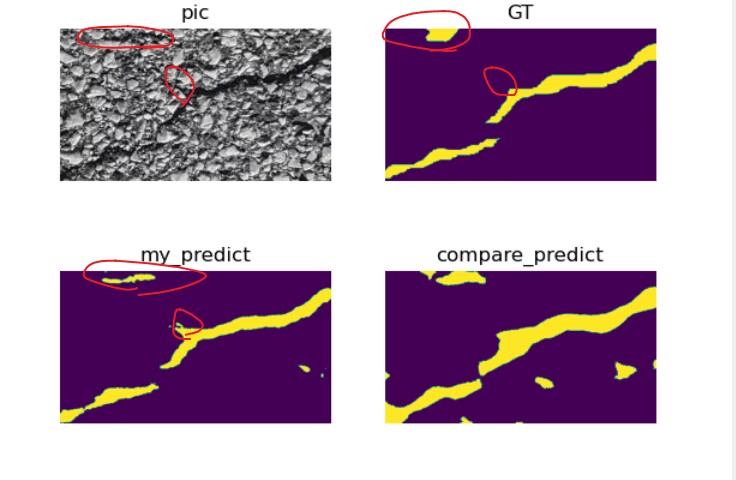
可以看到在IoU上，U-net比改造过的FPHB还要高出6个点。

下面是原图片，GT（Ground Truth），my\_predict(U-net),compare\_-predict(FPHB）



可以看出U-net的预测更加的精细，甚至在某些对某些裂纹图片的标记效果比人工标记的效果更好。例图如下：



2.8可改进方面

模型上，在Unet的基础上可以进一步将代码改进为U2net，即Unet++。

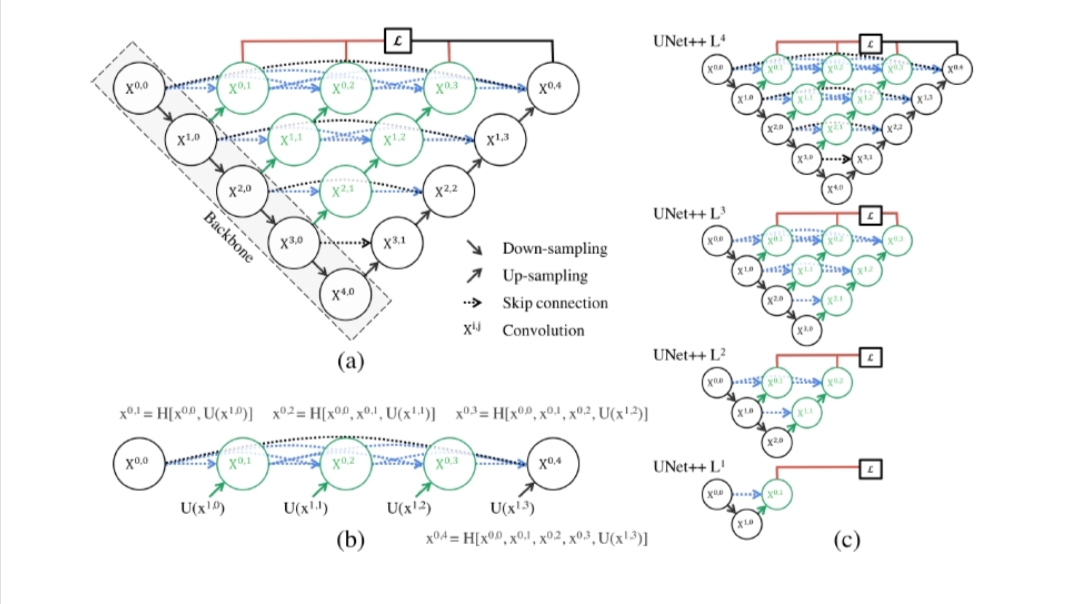
Unet存在的一些问题在于：

（1）最佳的深度是在先验上未知的，需要广泛的架构搜索或不同深度模型的低效率集成来测试

（2）跳接施加了不必要的限制性融合方案，仅在编码器和解码器子网的相同比例的特征图上强制融合。

Unet++模型能够更好的将不同层次的特征通过特征叠加的方式整合，相对于Unet使得融合时的特征图尺度差异更小。此外该模型通过不同深度的U-Net的有效集成来缓解未知的网络深度，这些U-Net可以部分共享一个编码器，并且可以通过深度监督（deep supervision）同时进行共同学习。

U2net模型架构如下图：



在数据预处理上，可以将所有的的图片通过将周围的位置进行填充，将所有的不同分辨率的图片的分辨率变得相同。如此操作避免了对图片的resize变换，能够更好的利用图片中的特征信息。

**3.系统管理模块**

3.1 开发环境：项目在springboot+VUE框架下进行开发。

3.2 用户子模块

3.2.1 注册登录

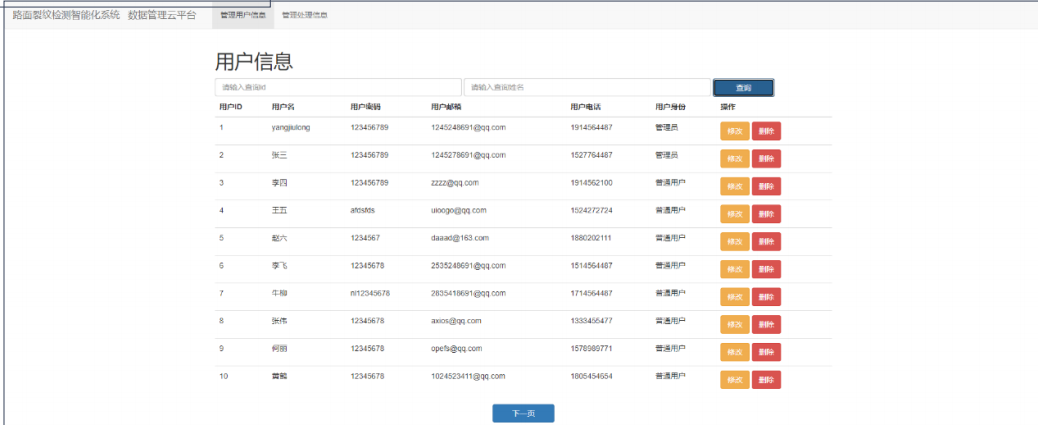
用户统一在app端注册，管理系统仅仅只有管理员可以登陆，普通用户并无权限。系统设置超级管理员一位，由其赋予其他管理员及权限。项目通过gateway网关进行统一的鉴权操作，防止没有管理员权限的人进入后台系统



3.2.2 权限管理

项目分级设置权限，高权限用户可以管理低权限的用户。超级管理员管理管理员，管理员管理普通用户，以进行管理员及用户信息的增删改查。同时超级管理员可将用户分配给相应的管理员管理。管理员只负责管理自身权限下的用户，一个管理员可以管理多个用户，获取所管理用户的检测记录和监测信息。该管理员只对所管理用户上传的数据可见，对其他用户上传的数据不可见。通过数据分析得到的统计数据，部分可以共享。





3.3 任务管理子模块

管理员通过该模块发布审批任务。对所分管用户进行任务的发布，并设置时间地点等信息。管理员可以对任务进行审核，查看用户任务的完成情况，并对任务实行更改通知。用户完成检测任务后，管理员能够获取对应用户的检测记录，查看用户上传的细节，如图像，地理位置等。

