经过了上一周的的讨论，我们对于上一周提出的两个问题进行了解决，并且对于论文中的introduction和method两部分进行了阅读和细节问题的探讨。  
首先是无引导上色结果的评价结果的量化，我们了解了两种人为设计的指标SSIM和PSNR。  
第一个指标是PSNR，它描述一张带有噪声的图片和一张干净的图片之间的差异，涉及到均方误差MSE,对于干净的图像I和噪声图像K，他们的尺寸都是M\*N，遍历所有像素值作差再平方求和，有了MSE之后就可以求PSNR了，这个括号里面MSE表示均方误差，上面的MAX表示图片的动态范围，也就是可能的最大像素值，如果我们的图像是8位的那么这个值就是255。当两张图片差异越小就意味着均方误差MSE就越小，信噪比PSNR就越大，图像的质量就越好。  
接下来第二个指标SSIM结构相似衡量指标用来衡量两张图片的相似程度 ，他衡量了三个性质，亮度、对比度和结构。其中亮度指的是所有像素值灰度的平均值，对于图像x的N个像素值求和取平均，同样的方式计算Uy，然后亮度对比函数就是可以表示为这个样子，当且仅当Ux=Uy的时候能够取到最大值1，也就是图像x，y亮度相同的时候函数值位1 ，为了防止像素值为0，造成分母为0或函数值为0，所以这里分子分母都加上了一个常数C。接下来是对比度函数（在统计学中，如果将一个样本的方差用样本大小n来除，得到的是样本方差的无偏估计。然而，当样本数量非常小的时候，无论该样本如何分布，其样本方差都会严重受到样本中的离群值影响。这可能导致对总体方差的估计产生偏差。在计算样本方差时，可以通过除以(n-1)来得到自由度校正的无偏估计。）。这里我们来求灰度的标准差来求对比度 ，同样分子分母加上常数来防止分母为0或者数值过小。最后是图像的结构对比，结构对比采用的是相关系数 ，用协方差除以方差的积来表示。所以最终SSIM的表达式是这个样子，把亮度、对比度和结构这三个特征乘在一起，其中α、β和γ来控制这三者的权重，且他们的值域在0-1之间，当且仅当两个图完全一样的时候各自取到最大值1。

接下来是计算图像的梯度方法。我们这里采用的是Sobel算子，他是计算横轴和纵轴两个方向的梯度，比如说横轴方向上我们用这个Gx的卷积核和原始图像相乘，比如我们右边这个3\*3的图像，我们要挨个点对它用卷积核进行计算，我们这里以P5在水平方向的梯度为例，它的计算是对应位置相乘然后加在一起，实际上就是P5右边这一列去减去左边这一列，同时还要注意系数，中间这一行它比较近，所以系数为2。如果右边的值和左边的值差距很大，那么P5就会比较大，所以P5就有很大概率是边界。相反如果两边值很接近，P5的值就会很小甚至为0，那它就不是一个边界。同样道理还能计算垂直方向的的梯度，它是用下面的这一行去减去上面的这一行。（在openCV中由于负数会被截断为0，所以通常要取个绝对值）然后x轴和y轴方向的梯度都有了，就把他们平方相加开根号。这就是p5这个点的方向梯度了。另外对于图像边缘点的梯度计算，由于边缘点他周围部分像素点是缺失的，不好直接计算，这种情况下可以利用边缘填充的方法，在图像的周围填充一圈像素点，可以进行0填充就是填上一圈0，也可以进行重复填充，就是讲原始数据的边缘值不断复制并填充到需要填充的区域。

在将controlnet之前我们要讲一下stable diffusion的基本原理，因为在论文中作者是基于stable diffusion这个大模型的基础上进行微调的。Stable diffusion相比于我们传统的扩散模型，他主要解决的diffusion模型的速度问题。Diffusion的缺点是在反向扩散过程中需要把完整尺寸的图片输入，这使得当图片尺寸以及time step t足够大时，Diffusion会非常的慢。而stable diffusion通过预训练好的自编码器来讲图像压缩到隐空间，这里面的伊普西龙就是编码器encoder，D就是解码器decoder来将隐空间的数据还原为原始大小的图像。通过编码器压缩后的图像在隐空间里面进行扩散。而在反向扩散过程中，他改进了去噪过程中的U-net结构引入了多头attention（在resnet之间添加attention），来处理文本向量（计算文本向量和当前图像的相关性）。图中每个黄色的小方块都代表一次注意力机制的使用，而每次使用注意力机制，就发生了一次图片信息和语义信息的耦合 。右边的conditioning，表示不同的条件输入，有语义生成图像，文字生成图像，语言描述生成图像还有图像生成图像。他们通过不同的方法转换成能参与运算的向量，比如文字就通过Text Encoder生成文字对应的embedding。然后这里的skip connection是指从Encoder阶段的每个编码器层复制一部分特征，然后将其与Decoder阶段的对应解码器层相连接。这样做的目的是帮助Decoder端恢复丢失的低级细节和上下文信息，提高模型的性能。然后switch组件的作用是，不是所有类型的condition都需要进行注意力计算的，也可以直接concat来融合。

由于stable diffusion不仅需要庞大的数据集和很多的资源才能训练出一个很好的模型，但是有时候一些特定的领域没有这么大的数据集并且只有有限的资源，就需要利用已经训练好的stable diffusion模型在上面微调。所以就有了controlnet的小模型来控制大模型。像右图中的这样将stable diffusion分为locked copy和trainable copy，locked copy这部分是在训练过程中参数是不会调整的，而trainable copy这部分是训练过程中去调整它的参数。作者的思路就是用locked copy这部分保留网络原来的能力，而用trainable copy这部分去从小样本中学习怎么控制生成。这里面还有一个零卷积层，通过它来将两个部分连起来，零卷积层其实就是一个权重和偏差为0的卷积层。这里是指训练开始的时候为0 ，训练之后经过一次迭代就不是0了开始进行更新了。这里加0卷积层的目的就是为了保证训练的一开始，模型的输入输出和我不加controlnet的结果是一样的。所以就是原模型的输出加上contronet的输出就得到了我这个模型的输出。通过这种方法，他就可以比你从头开始训练这个网络是更快的。然后这就是controlnet完整的结构图，左边这就是stable diffusion不断降采样，然后升采样，中间相同的上采用或下采样的层之间连起来。右边就是contronet新加的部分，encoder部分都是copy左边的模型结构和参数，但是decoder部分他是没有的，然后通过0卷积层去吧encoder这部分输出和左边decoder部分加起来。（每个block里面是resnet加上transformer的结构），作者针对不同的数据集还给出了不同的改进建议，对于比较小的数据集可以只留下连接这个middle block的0卷积层，其他的都删去，就保证了训练的高效，减少了参数的更新。而对于大数据集，可以在训练中途的时候讲stable diffusion后半部分的decoder部分也进行更新，就是不lock了，就可以学的更好。