见解：

上色的过程大致应该是我们输入一张线稿图给controlnet，controlnet通过边缘检测理解图上包含了什么信息，然后在边缘的引导下对线稿进行上色，因为我们的stable diffusion模型一开始是在大数据集上训练的，所以生成出来的图可能比较大众化，有时候我们并不满意这个结果，想要生成特定颜色、特定风格的图片，我们可以将我们的prompts加入进去，通过clip模型来达到自然语言与图像的匹配，并在边缘的约束下将图像创作到其所在的区域，最后达到我们想要的上色效果。

Ai上色与我们一般意义上的人工上色有所区别，人工上色是我们看到一张线稿，理解线稿上的内容并决定线稿每个区域应该填充的颜色，而ai上色是理解线稿上的内容，再由ai根据prompts自由创作和我们的线稿图内容类似的图像，并通过controlnet的canny模型检测线稿的边缘信息来决定生成图像的位置布局，这是一个从无到有的过程。

假设我们不用controlnet，单纯用sd模型，将我们的线稿图输入进去，我们发现模型生成出来的图片只是形状与原图片相似，可以说完完全全是一张新的图片，因为他并没有理解我们图像中具体包含了什么内容，只是根据我们的prompts将原图像魔改成提示词中所要实现的效果。

而当我们加入controlnet之后，controlnet会去检测图像的边缘信息，理解图像每个部分包含的含义，并将此作为一种特殊的prompts传递给sd模型，所以在无参考提示的情况下，sd模型仍能够生成我们所期望的图像，在生成图像的过程中，模型会受到controlnet的引导，使得图像生成在线稿约束的区域之中，总而言之，controlnet在线稿上色中承担了两个作用，一个是提取图像的特征，作为模型的prompts，另一个是控制图像的布局，使其呈现上色的效果。

sd的图生图+ControlNet合起来用就好比，你给画师两张图。让画师参考ControlNet的图，然后魔改给sd图生图用的图。画师就先理解ControlNet图里面的各个部分有什么东西，然后尽力在你给的sd图生图里面的图加各种线条魔改，努力魔改成ControlNet里面的图的各个部分的东西，也就是这时也不理解sd图生图里面有什么东西

sd的图生图是在你给的图片的基础上根据你给的提示词继续降噪扩散。而不是从随机噪声开始扩散

sd的图生图就好比，你给了一张图给画师，让画师画成指定内容。然后画师就在你给的那张图上加各种线条魔改成最终图像，画师并不认识你给的图中的每个部分是什么，只知道你给的提示词应该要有什么效果

controlNet就相当于，你给了一张图给画师，让画师画成指定内容。画师就会先去理解你给的图片里面有什么东西，根据理解的东西进行上色处理与细节补全

对这些区域的判断是ControlNet理解的

是ControlNet模型训练的不是sd模型

prompt是给sd（的Clip层）的，或者是同时给sd和ControlNet（主要还是前者，后者1.1版本才比较多）

你可以认为纯sd的图生图是在你给的参考图的基础上继续降噪扩散，而不是从随机噪声开始由cn引导扩散

上色的过程大致应该是我们输入一张线稿图给stable diffusion模型，模型会先去理解这张图描绘了什么，他用到了clip模型，来达到图片和我们自然语言的匹配的目的，然后模型会运用大数据集中学习出来的能力对线稿进行上色，生成出来的图可能比较大众化，有时候我们并不满意这个结果，就要对模型进行fine-tune，controlnet就是其中的一种方法，它将模型分为trainable copy和locked copy，我们将我们的条件加入到controlnet中去，他会在trainable copy上训练参数来适应我们的要求，最终将输出与我们预训练的模型的输出进行叠加，从而达到我们想要的上色效果。

就是clip是理解你的提示词的不是你输入的原始图片的

不要把预处理和ControlNet放在一起想，预处理器是将不是线稿的图转化为线稿，我们目标本来就是给线稿上色所以不需要用预处理器。ControlNet的canny模型就是控制sd生成图像的边缘的。  
也就是说，canny预处理：把图转为线稿  
ControlNet的Canny模型：控制sd参考线稿生成图

ControlNet玩法太多了，我说几个我玩的比较多的  
1.控制构图，纯sd提示词是一个一个词，所以话多人图的时候很难将每个提示词对应到每个人上，特征容易串，用ControlNet控制，canny+openpose+seg三重控制构图很稳  
2.整理草稿，拉低分辨率用ControlNet的canny模型上色，然后再重新对结果提取线稿再用一次canny，可以清理手绘草稿的一些杂乱的线条  
3.多帧视频渲染，将视频的上一帧图像输入ControlNet的canny+depth+normal模型，生成下一帧，依次类推，根据提示词一帧帧的生成动画  
4.局部重绘：将部分地方有瑕疵的图（例如去马赛克）输入ControlNet的canny+inpaint模型，局部重绘  
5.画风迁移：使用ControlNet的IP2P+Shuffle+Canny模型，配合不同风格的sd底模，可以将同一张图的画风变为sd底模的画风保持物品不变  
6.三次元图片二次元化：ControlNet的tile模型+canny模型配合二次元sd底模可以将三次元图转二次元。当然同样可以用这样创造各种类同人图。