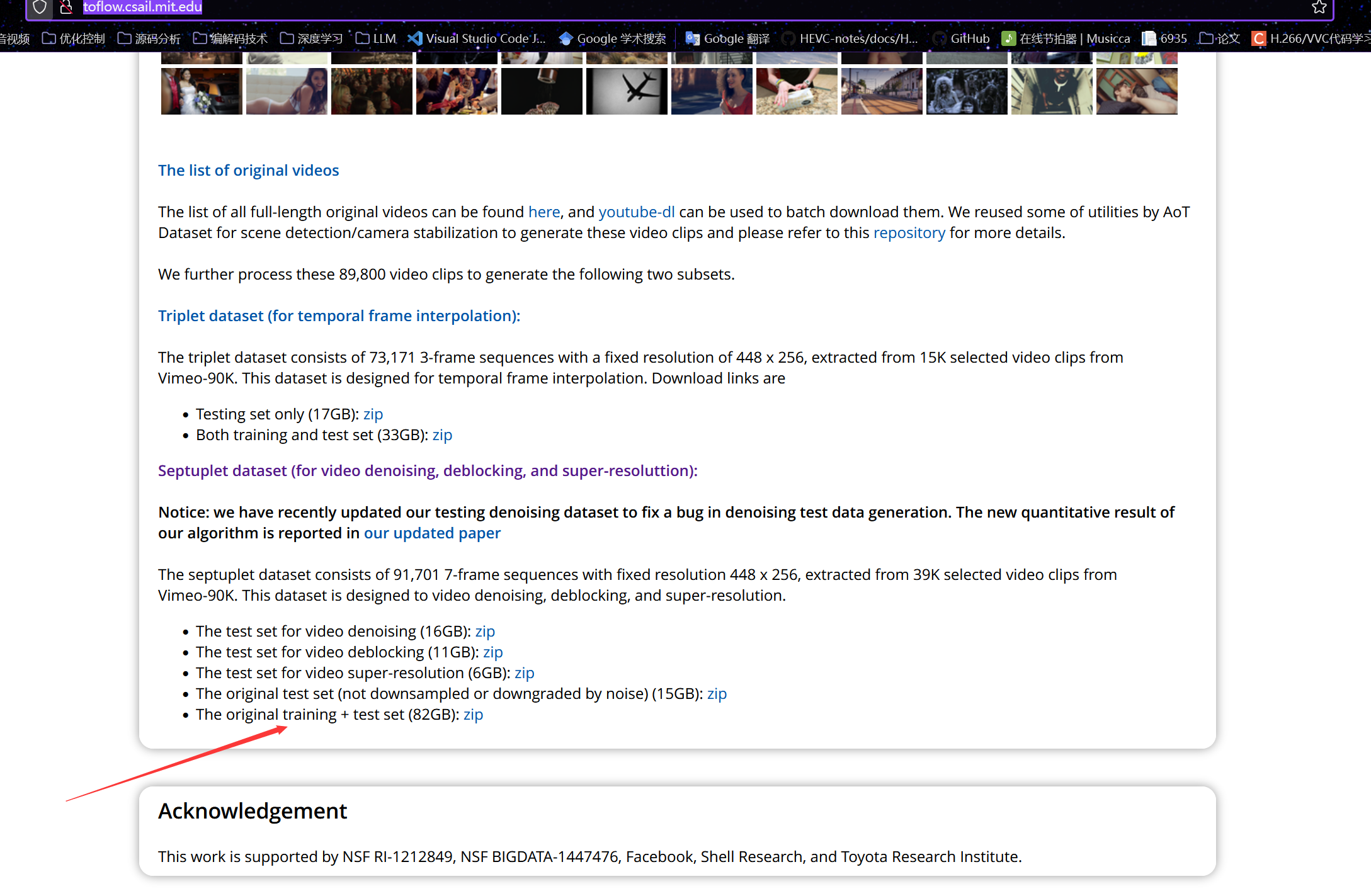
**视频隐写网络**

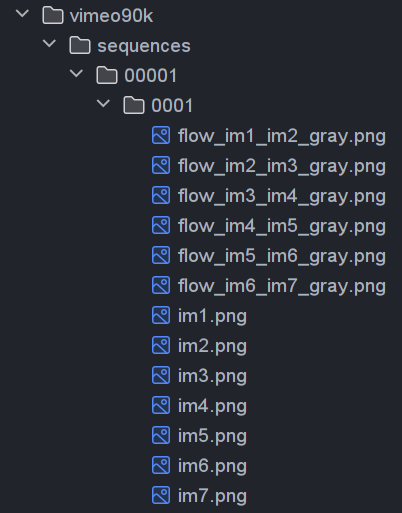
数据集

1.vimeo90k <http://toflow.csail.mit.edu/>



2.基于vimeo90k，用mmflow跑光流数据集 <https://github.com/open-mmlab/mmflow>

代码打包在vimeo90k\_image\_demo.py，功能是把vimeo90k每一组图片里，每两张png输入mmflow网络，跑出一张光流图flo，将光流图flo转换为灰度后，保存在vimeo90k文夹下，结构如下图，im1~7.PNG是vimeo90k的原始数据集，gray.png是脚本跑出来的光流灰度数据集。



## 环境

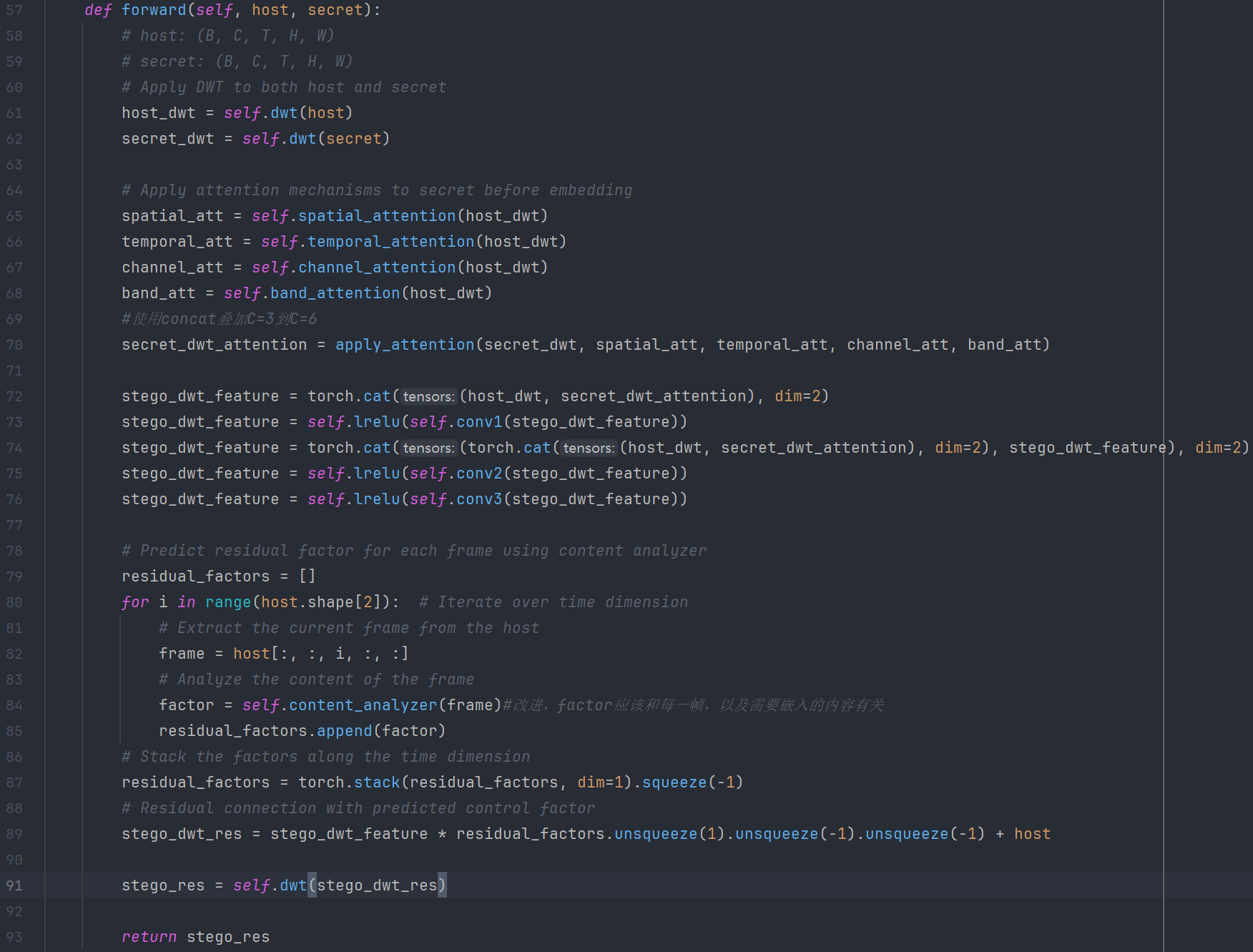
使用mmflow的环境

## 系统功能

视频隐写，输入7张连续的PNG（vimeo90k里是每7张作为一个视频片段），在PNG中嵌入秘密信息。具体的，训练时，生成随机0，1比特，叠加到PNG中，得到隐写后的PNG，对隐写后的PNG施加失真，以模拟视频编码过程中的失真。

## 训练流程

1. 数据集加载，从vimeo90k中加载im1~7.png和对应的光流灰度图。
2. 训练过程，以encoder类为例，代码打包在optical\_Vstego.py



Host为加载的im1~7.png，（1,C,T,H,W）=(1,3,7,H,W)表示1组三通道，7张png。

对host中每一张每一通道都进行dwt变换，变换后尺寸为（4，C,T,H/2,W/2）,得到host\_dwt

Secrect为随机生成的01数据，尺寸为（1，1，T,H,W），进行dwt变换后尺寸为（4，1，T，H,W）,得到secret\_dwt

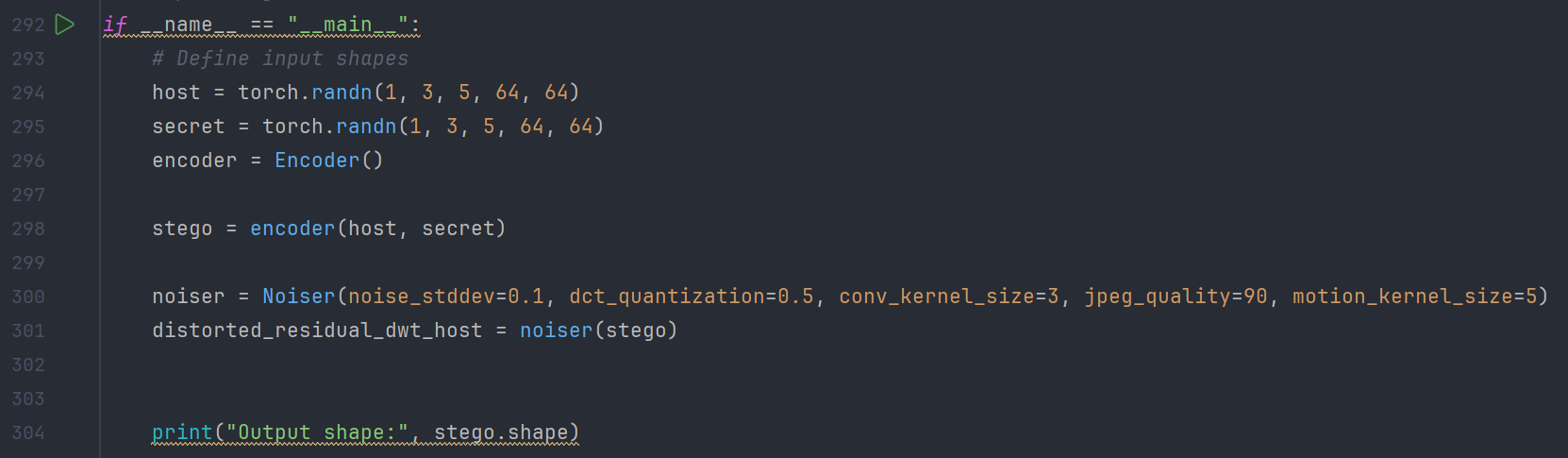
后续进行空间注意力提取，时间注意力，通道注意力提取，使用这些注意力后调整secret\_dwt,特征提取，具体怎么卷积你看着做，我代码里的可能跑不通。

最后进行host\_dwt和secrect\_dwt的融合，host\_dwt+λ\*secrect\_dwt，lambda是生成的，根据host\_dwt生成。

融合后，进行dwt的逆变换iwt，重新转换为7张png

至此，隐写结束。

1. 叠加噪声

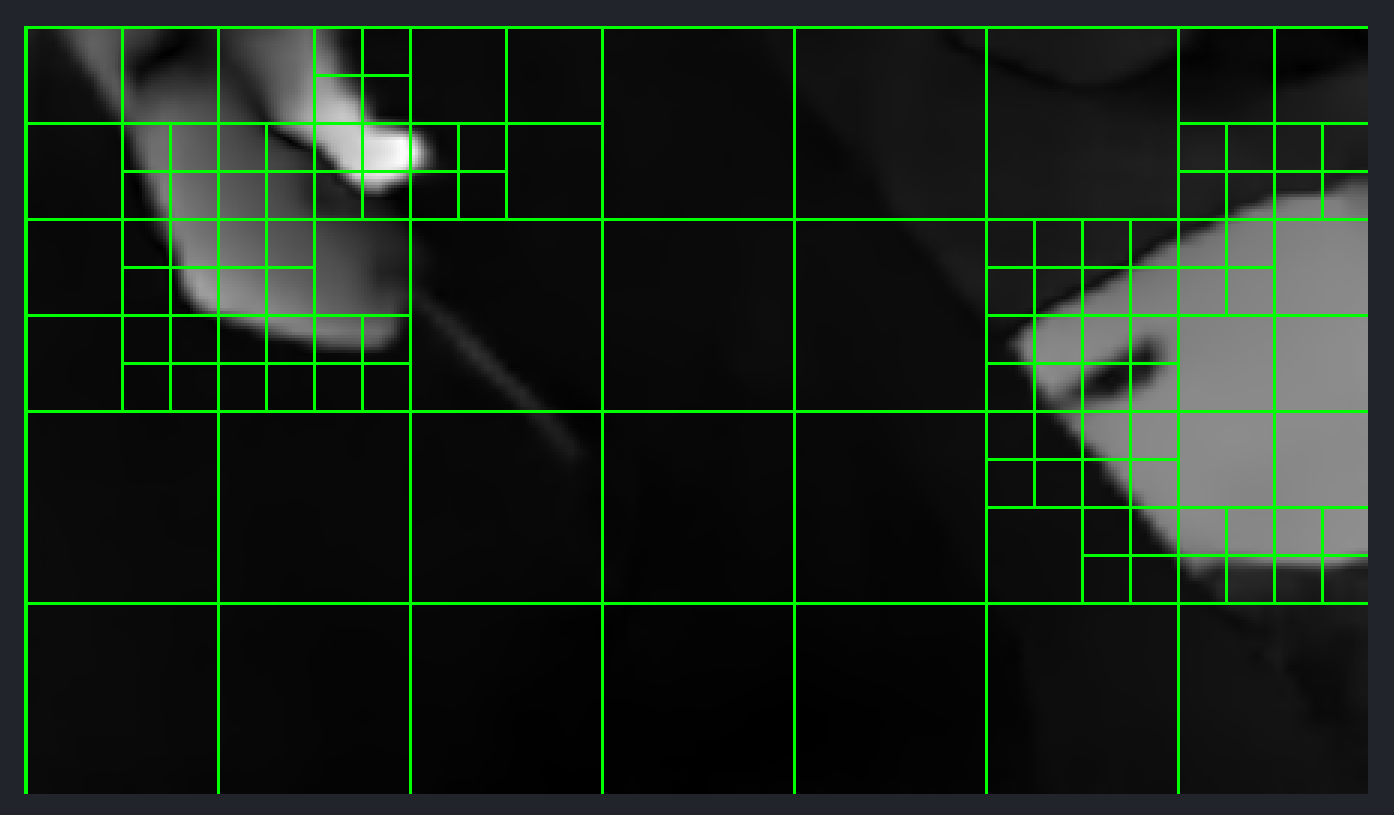
对于隐写完成的每张png叠加噪声，

Noiser类里面定义了随机噪声，jepg压缩等，对隐写完成的png叠加噪声，注意都是以块为单位叠加噪声。

下图是对光流灰度进行块划分后的结果，进行块划分的代码在flow\_grid\_flow.py。以块为单位添加噪声。并且噪声根据灰度图块内亮度的均值来进行调整，越亮的块叠加的噪声越大，比方增大随机噪声的方差，增大jpeg的压缩程度。

得到添加噪声后结果

下图是flow\_grid\_flow.py分块划分后的示意图



Decoder

对添加噪声后png的进行解码，解出encoder过程中写入的信息secret，整个过程就是encoder的逆过程。

损失函数

计算原始png和隐写后(写入secrect后，还没叠加噪声的PNG)PNG的psnr，（类似两张图片的相似度）

计算提取出来的secret和原本的secret的相似度，这里采用逐比特对比，看正确的比特书的比例。

把两个相加共同作为损失，反向传播。