**control.py**

run control.py to start the program

**embed.py**

**watermark pro-processing**

The watermark pictrue is fingerprint.jpg.It will be processed by Median filter, gray scale, black and white binarization.Then we get the picture named embedfinger.jpg

**host pro-processing**

The host picture is host.jpg.It will be processed by RGB-to-YUV,uint8-to-float32.Because we will DCT on the Y dimension which means the value of brightness later,and DCT need float32.

**watermark embeds into the host**

* Divide host into 8x8 blocks.
* Calculate the average number of fingerprint pixels to be stored for each 8x8 block.You can see output r is it.
* Each block on the Y dimension DCT for the airspace change into frequency domain.Then we discuss the operation to 8x8 blocks in frequency + domain.
* In a 8x8 block, the unit cell,and its symmetric unit cell of the center,become a pair,named r1,r2
* Each pair of relationships (r1>r2,r1<r2) is used to record the 0-black and 255-white of the fingerprint pixels to be stored.
* Embed the average number of fingerprint pixels into mid-frequency cell of the 8x8 block.
* Each 8x8 block on the Y dimension inverse DCT for frequency domain change into the airspace.The finished product is finishwm0.jpg(finish0~5.jpg are different intensity of JPEG compression,finishwm0.jpg is best quality,finishwm5.jpg is worst quality.)

**extract.py**

Just 8x8 DCT and find embeded location,then sure the average number of pairs' relationship(r1>r2 is 0-black and r1<r2 is 255-white).Constitute Y dimension of original watermark named 0extractfinger.jpg(0~5)

**evaluate.py**

Calculate PSNR and observe products from different JPEG compresstion.