**Watermarking-based image authentication with recovery capability using the block-basedhalf-toning technique (2011) by Luis Rosales-Roldan**:

This article introduces two algorithms: WIA-IWT (watermarking-based image authentication use integer wavelet transform), WIA-DCT (watermarking-based image authentication use discrete cosine transform) for image authentication, localization, and recovery.

Both algorithms contain three stages: self-embedding stage, authentication stage, and recovery stage. In the embedding stage, we have two different processes. First process is to generate a watermark sequence. Watermark sequence is generated from the down-sampled original image with the half size of its height and width. Once the original image is down-sampled, they use the error diffusion half-toning method to generate the halftone image. Then, they use permutation algorithm for halftone image with the user secret key. Finally, they generate the watermark sequence. The embedding process of those two algorithms are a little bit different. WIA-IWT will decompose the image to four sub-bands: LL, LH, HL, and HH. Then, they put the watermark sequence into LL area. While WIA-DCT method will divide original image into 8\*8 pixels. Watermark sequence will divide into 16bits for each blocks and they embedded them into the middle frequency range. Finally, using inverse IWT/DCT transform to get watermarked image.

In authentication stage, we extract the watermark first, and then do authentication process using the extracted watermark sequence. In extracting watermark process, they extract watermark bits w~ n by using quantization-based method and reorder it by using user secret key. After that, we get a half-tone version of the original gray-scale image which is image B. They also generate a halftone image from down-sampled suspicious watermarked image and re-convert into aa gray scale image which is image A. Then compare A and B. Inverse half toning method is important in this process.

In the recovery page, if the image is tampered, they use the halftone version, down-sampled watermarked image and extracted halftone image C as the input data. They use MLP-based inverse halftoning method for recovery. First, they use down-sampled suspicious image and its halftone version to train MLP by using backpropagation algorithm. After that use extracted half tone version and tampered area determination as input in MLP application to get the recovery image.

**A hierarchical digital watermarking method for image tamper detection and recovery (2005)**

by Phen Lan Lin

The idea of this article is to create a hierarchical structure for tamper localization. It is also effective since it only requires a secret key and a public chaotic mixing algorithm. The tamper detection will through a simple three-level, block-based inspection of in-block information only. The recovery of the tamper block relies on its feature information hidden in another block. The whole scheme can also be divided into three stages: watermark embedding, tamper detection, and tampered image recovery.

In watermark-embedding process, there are two processes: preparation, block watermark embedding.

First, preparing image with M\*M pixels and number of gray levels is 256. Then, they prepared a block mapping sequence for watermark embedding. Here is an example: A->B->C->D->…->A. The most important feature of A will be embedded in B, and the most important feature of B will embed in C.

Second, dividing each block (4\*4 pixel) into 2\*2 pixels. Each sub-block is a 3 tuple with (v,p,r). v and p, are authentication watermark, r is recovery watermark. R is corresponded with block A to block B. First, we set two LSBs of each pixel with in the block to zero and get the average intensity of a sub-block A which is As. Then, we get intensity r of as from six MSBs. Embed all 3-tuple watermark onto two LSBs of each pixel with in Bs.

During the tamper detection process, they did the same thing that divide test image into 4\*4 pixels. Each block is denoted as B’. We get the average intensity which is avg\_B’. Then, we perform 3-level detection. In level 1, we treat 2\*2 sub-block within one block, we extract v and p and then compare p with the new p’ to check if there is an error. In level-2 detection, we treat 4\*4 block as one unit. If sub-block doesn’t have error, it is valid; otherwise, mark it erroneous. Finally, we check the block by extending the inspection view to its 3\*3 neighborhood. If there are five or more errors blocks in its 3\*3 block neighborhood, mark it as erroneous. Level 4 detection is only used for resisting against VQ attack.

After detection stage, all the blocks are marked as either valid or invalid. We only recover those block with error which denote as block B, and the intensity in block B is block C. First, we find block C, if block C is erroneous, skip the recovery. If block C is valid, obtain the 6-bit-intensity of each subblock within block B by firstly extracting the two LSBs from each pixel in the corresponding sub-block within block C then discarding the bits v and p. Pad the 6-bit-intensity with two 0s to the end and replace the intensity of each pixel within the sub-block with this new 8-bit intensity. Repeat steps for all sub-blocks within block B and mark block B is valid.

**An efficient watermarking technique for tamper detection and localization of medical images (2018) By Solihah Gull**

This article intro­­duce a new technology of tamper detection and localization for medical image. They suggest that divide a cover image into 4\*4 pixels blocks. Each block divide into two equal blocks. The upper half block contains the information of tamper localization. The lower half block will contain the tamper detection information.

The process of embedded algorithm is the following steps: 1. Dividing original image into 4\*4-pixel block which denote as Bx. 2. Letting all the LSB of every pixel to zero, we call it Ba. 3, computing the arithmetic mean of Ba. 4, dividing Bx into two 4\*2 blocks: LHB and UHB. 5, arithmetic mean in step 3 XOR the watermark pixel. 6, converting the result into the bit stream and embedded into LHB. 7, convert the mean value from step 4 and embedded into LSB of UHB.

The process of extraction algorithm is the following steps: 1, dividing watermarked image into 4\*4 blocks. 2, set the first two LSB of each block to zero and compute the mean of each block. 3, using XOR operation between mean and extract mean bits from LHB. Then, we get the extracted watermark. 4, compare the computed mean with one extracted from UHB. If they matched, no tamper; otherwise, it is a tampered block.

**Digital Watermarking for Telltale Tamper Proofing and Authentication (1999) – By Deepa Kundur.**

This article presents the fragile watermarking approach which embeds a watermark in the discrete wavelet domain of the image by quantizing the corresponding coefficients.

There are three main stages of the watermark embedding procedure. In the first stage, they compute the Lth-level discrete wavelet decomposition of the host image the generate a sequence of three L detail images which corresponds to horizontal, vertical, and diagonal details at each of the L resolution levels.

**Secure and Robust Fragile Watermarking Scheme for Medical Images (2019) by ABDULAZIZ SHEHAB**

Watermark embedding process: There are two directions in this process: first is to get the authentication bits, the second part is to get the recovery bits.

For authentication bits, we first divide image to 4\*4 image blocks and LSB for all blocks are set as zero. Then, using SVD operation to calculate the U, S, V matrices as the authentication bits, which are singular matrix (left and right), and singular matrix. They narrow the range of authentication bits to 12 bits.

For recovery bits, they first divide each 4\*4 block into 2\*2 blocks and calculate the average value of each 4\*4 blocks from 2\*2 block. After that, we only get the first five MSB from each block and put them together as 20 bits recovery bits. After that, we combine 12 bits and 20 bits together as our watermark and put them into the host image by replacing the last two LSBs of each 4\*4 block.

During the extraction process, the idea is nearly the same. We still need to divide watermark image into 4\*4 block division and we also have two direction.

The first direction is image authentication. First, we extract the last two bits from every block and figure out the authentication bits, which is 12 bits MSB. Then, we set the image block with zero LSB and calculate the U, S, V by using SVD operation. After that, we compare the extract authentication bits and the authentication bits we got from last step. If they match, mark the block valid; otherwise, it is invalid.

If it is invalid, we need to recover the image. In this step, we just need to use 20 bits self-recovery bits for image recovery.

**Image tamper detection and self-recovery using multiple median watermarking (2019) by Vishal Rajput**

The idea of the methodology is that inserting multiple watermarks into the host so that the chance of recovery will be increased.

Embedding process:

1. Divide host image to 4 parts (w, x, y, z)
2. Save the half-size copy of the host image into another variable.
3. Convert step 2 result into binary value and take the 3 MSB to form a 256 \* 256 matrix
4. Use the 4 different random order of size 256 \* 256 to save the matrix from step 3 and put them into LSB of the 4 different parts w, x, y, z.

Recovery process:

1. Extracting information from 4 LSB’s of 4 images. Using 4 recovered image to cover a median image.
2. Passing median image through filter to remove noise.
3. Comparing median image or any part of image with the tampered image to find the tampered region.
4. Using median image or any other watermarked image to recover it.

Self-Embedding Authentication Watermarking with Effective Tampered Location Detection and High-Quality Image Recovery (2019) by Chin-feng Lee

This article introduces two different detection which are block-wise and pixel-wise. Both of them contains two process. The first process is the watermark generator. The second process is the authentication and recovery.

For block-wise method:

Watermark generation:  
We first divide H\*W size image to N number of equivalent m\*n block. Second, we calculate the mean of each block and use that mean value as the recovery data. Third, we generate mapping block by using secret key, and then we use the mean value of the mapping block BM as the recovery data. Forth, we use two equation to generate authentication bits: A. Then, we combine authentication and recovery data together as the watermark. Finally, embedding the watermark into LSB of each block using LSB replacement method.

Tamper detection and image recovery:

First, we divide watermark image into m\*n blocks, we recalculate the authentication data by using the same method above to get the A’. Third, we extract the watermark data from each block and get the authentication data, recovery data. Compare the A’ and A to determine the tampered area. Forth, get the mapping block by using the same method. Fifth, if the mapping block is no tampered, we can see the recovery data is not tampered, otherwise, we have to repair the recovery data by using surrounding pixel values and using interpolation method. Finally, enlarge the recovered image to original size.

For pixel-wise method:

The reason why two copies because the survive chance will increased