

Multimedia Data Security - Competition Rules

What is the competition about?

The Multimedia Data Security competition is essentially a group activity that allows applying already learned concepts in a fun way. The objectives are two:

- Applying an **embedding** strategy that is both **robust and unperceivable**
- **Attacking** the watermarked information of other groups while **preserving image quality**

NOTE: In the following document, each part is discussed thoroughly. If you still have doubts, the teaching assistants are more than happy to answer your questions, so make sure everything is as clear as possible!

Scheduling

Deadlines:

Wednesday, 02th October 11.59 PM
Sunday, 27th October 11.59 PM

GROUP FORMATION
EMBEDDING, DETECTION, AT-
TACK and ROC CODE SUBMIS-
SION

Events:

Monday, 4th November 8.30 AM - 12.30 PM
Wednesday 6st November 11.30 AM - 13.30 PM

COMPETITION
GROUPS PRESENTATIONS

Specific labs will be devoted to code development.

Group Formation

All participating students are divided into groups of **3/4 people**. Each group must select

- a **spokesperson**, that will take care of the communication with the teaching assistant
- a **nickname** that only your group will know. Group nicknames must not contain spaces/symbols/-capital letters.

Once everything is ready, the first job of the spokesperson is to send an e-mail to the teaching assistants and the professor with the name of the group and the name, surname and e-mail (preferably academic address) of each member of the team.

The object of the mail **MUST BE**: [CTM 2024] - *groupname*.

This must be done by the following **deadline**: **02th October at 11.59 PM**.

After the deadline for registration, you'll receive an e-mail providing:

- A randomly generated **watermark** which will be strictly associated with your group. All information about the watermark is given in the *Additional Information* section at the end of this document.
- A randomly generated **password** that you will need to access the website of the competition (so keep it safe and secret).

Running Code Submission

Each group should work on its strategy. You can work also before the watermark is given to you: just create a dummy watermark (you can see the structure in the *Additional Information* section) to test your code with.

By **27th October at 11.59PM** you will be asked to deliver a working code. This is just to avoid you having problems the day of the competition, and have the backup of a working solution. You are asked to deliver also the ROC-curve code so that we can check how you set the threshold. You can still fine-tune your solution after this deadline.

Meaning of Fine-tuning: adjust parameters and embedding locations. Other things are forbidden. For more details see the section “How to prepare your code”.

Competition

The competition will take place on **4th November**, please **be ready to start at 8.30 AM**.

The website will be your main tool to upload and download the materials.

Only one PC per group can access the website of the challenge.

Groups can bring at most **3** laptops. You CANNOT use any server.

The competition is divided in two phases: defense phase and attack phase. Each phase is strictly timed, so be ready!

Important note: If all of you try to access the website at the last minute of the competition, you might experience problems in uploading content.

Defense phase → 8.30 AM - 10 AM

During the defense phase, you will have to

1. Download the challenge's images (three 512×512 grayscale images)
2. Download the WPSNR code (if needed, as it will be the same code provided during laboratories)
3. Use your embedding strategy to insert your watermark in each of the three images
4. Upload the embedded images to the website. You can do this multiple times, each time you do it a score will be assigned to you based on the quality of the embedding (see Scoring Information for more details).
5. Upload the code of your detection strategy (other groups will use it during the attack phase)

During this phase, you can tune some parameters of your embedding **AND** detection strategy. Make sure that you correct everything IN BOTH CODES before uploading them on the challenge website.

Remember to check if your detection code works properly!

Attack phase → 10.15 AM - 12.30 AM

The second phase of the competition consists in targeting the other groups and attacking their images following these steps:

1. download other groups images from the list that will appear on the website
2. performs some image processing attack(s) on the images watermarked by other groups to remove the watermark while keeping image quality as high as possible.

Important note: the list of attacks that you are free to use, both singularly or combining them, or applying them globally or locally, is reported in the *Additional Information* section. **Note:** no external software (e.g.: Photoshop, Paint, GIMP, ...) are allowed.

3. uploads the attacked images on the website, indicating which group was attacked. If you perform multiple successful attacks on a group's images, just upload the case in which you have the highest WPSNR and the watermark was destroyed!

NOTE: using the original image to localize the watermark is NOT allowed.

Verification phase

After the competition you are required to send a log of your attacks (use excel). The required structure is the following:

Image	Group	WPSNR	Attack(s) with parameters
lena	attackedGroup1	37	JPEG QF=90, Median 5×5
...

We will use this log file to check your results. Please, be aware that cheating will be penalized. We will also check any other problems that might arise during the competition (code not following the rules) and update scores accordingly.

Presentation

Each group will have to present its approach on **6th November** as part of the exam: each group member has to be present, take part in the presentation, and answer possible questions. Once we will be aware of the number of groups participating we will provide more details about the timing of the presentation. The presentation should include the following elements:

- defense strategy
- attack strategy
- results

How to prepare your code

It is very important that your code strictly follows the rules
: penalties will be given to those not following the rules. [See the *Additional Information* section for more details]

Both the embedding, ROC (used for threshold estimation), detection, attack and a demo code (showing how to run each of them) must be sent for a final check and evaluation before the following date **27th October at 11.59 PM**. Only code fine-tuning will be allowed after that date. If during the final check the assistant finds errors or uncertain steps, an e-mail will be sent to the corresponding group, asking for clarifications and corrections to avoid mistakes and cheating during the day of the competition.

Embedding

Make sure to use `opencv` library to read, write and generally handle images. All the images will be in BMP format. During the competition, you will be provided with three grayscale images of fixed resolution 512x512. As explained later, points are assigned on basis of the **WPSNR value** resulting from a comparison between the original image and its corresponding watermarked version: higher the WPSNR higher the score. [See the *Additional Information* section for more details]

The code for the embedding **MUST follow** this structure:

```
def embedding(input1, input2):  
    '''  
    YOUR CODE  
    '''  
    return output1
```

- `input1` corresponds to the **string** of the name of the **original** image
- `input2` corresponds to the **string** of the name of the **watermark**
- `output1`: is the watermarked image

NOTE: Your code must NOT print anything, neither open interactive pages or windows or similar.

Threshold estimation

Once you gathered what should be the embedded watermark, the next step is to understand if the extracted data corresponds to the original watermark or not. For this competition, a **similarity evaluation** must be performed. For this purpose, the detection strategy is prepared in two phases: **threshold computation** and **comparison**. [See detection section for the comparison]

The threshold computation must be performed **once** and not in the actual detection code that you will deliver on the competition day. Since it is unrealistic to have image-specific thresholds **only one threshold** value must be selected and used **for all the images**. The code to compute the threshold **should not be included in the detection algorithm**.

Once you completed your watermark extraction strategy, the similarity threshold **can** be estimated by modifying the code of Lab4, topic ROC-curves example 2, as follows:

1. Embed your watermark $W_{original}$ to a set of images
2. In a loop, attack one by one these images (with random attacks or the strategy you prefer)
3. Extract the watermark with your planned technique $W_{extracted}$
4. Compute $\text{sim}(W_{original}, W_{extracted})$ and append it in the *scores* array and the value 1 in the *labels* array. These values will correspond to the true positive hypothesis.
5. Generate a random watermark W_{random} and compute $\text{sim}(W_{random}, W_{extracted})$ to append it in the *scores* array and the value 0 in the *labels* array. These values will correspond to the true negative hypothesis.
6. with *scores* and *labels*, generate the ROC and choose the best threshold τ corresponding to a False Positive Rate $FPR \in [0, 0.1]$.

Take note of the τ value: it will be used to assess if the data extracted with your strategy from a filtered/attacked image still corresponds to the original watermark. **This code has to be sent within the 27th of October at 11.59 PM**

Detection

The detection of a watermark is strictly related to your chosen embedding strategy. For the purposes of our competition, the watermark detection follows a **non-blind** strategy.

The detection function must be a **single file** named `detection_groupname.py`, no external functions are allowed (except those of the WPSNR).

The function will make use of the **WPSNR** code seen during the laboratory sessions.

YOU MUST NOT HAVE RELATIVE PATH OR EXTERNAL SCRIPTS CALLED IN THIS FUNCTION

You need to make sure that **input and output values are correct** and follow this structure:

```
def detection(input1, input2, input3):  
    '''  
    YOUR CODE  
    '''  
    return output1, output2
```

- `input1` corresponds to the **string** of the name of the **original** image
- `input2` corresponds to the **string** of the name of the **watermarked** image
- `input3` corresponds to the **string** of the name of the **attacked** image
- `output1`: if the attacked image contains the watermark it is equal to **1** , otherwise it is equal to **0**
- `output2` corresponds to the **WPSNR** value between the **watermarked** and the **attacked** image.

To assess whether an attack is successful or not, you must compare the watermark extracted from the watermarked image (i.e.: $W_{extracted}$) and one extracted from the attacked image (i.e.: $W_{attacked}$). Therefore, the detection code should not rely on the watermark file nor have the watermark hard-coded. Hash or derivation of the watermark (like SVD) can be hardcoded. If the similarity between the two values (i.e. $\text{sim}(W_{extracted}, W_{attacked})$), **is equal or above** the previously calculated threshold τ , then the watermark is assumed to be present and the attack is considered failed. An **attack is considered successful** if

- the similarity is below the threshold τ (i.e., `output1 = 0`)
- the $\text{WPSNR} \geq 35$ [dB] (i.e., `output2 \geq 35`).

Summing up:

- Read all the images inside the function using the three input paths!
- Don't read the original watermark: extract it!
- Attacks are considered successful only if the watermark is destroyed and $\text{WPSNR} \geq 35$ [dB]
- Attacks are considered failed if the watermark is present or, if destroyed, with $\text{WPSNR} < 35$ [dB]

Important note: Make sure that if you provide the original image also as attacked input to your detection function, the watermark is not present.

Final remark: your code must complete the detection within 5 seconds and must not open any pop-up windows or print anything on the screen.

Attack

During the attacking phase, you are asked to work using image processing techniques.

You are only allowed to use a limited list of attacks, that you can tune, combine and localize to destroy other groups watermarks. **Permitted attacks** for the competition are:

- AWGN
- Blurring
- Sharpening
- JPEG Compression
- Resizing
- Median filtering

Using the original image to localize these attacks **IS NOT ALLOWED**. The code **CAN follow** this structure.

```
def attacks(input1, attack_name, param_array):  
    '''  
    YOUR CODE  
    '''  
    return output1
```

- `input1` corresponds to the **string** of the name of the **watermarked** image
- `attack_name` can be a single string or an array of string containing the name of the attacks you are using. The attacks name **CAN BE** 'jpeg', 'awgn', 'blur', 'sharp', 'jpeg', 'resize' and 'median'.
- `param_array` is an array of parameters to run the attacks
- `output1`: is the **attacked watermarked image**

Provide a README file explaining how to use this script when you sent it to the professors.

Additional Information

Watermark

Each group will be assigned a specific watermark containing the assigned mark (a 1024 NumPy array of zeros and ones), that will be used on the day of the competition.

Naming convention

Assuming that **groupA** is your group name, the embedded images must be named

`groupA_imageName.bmp`

Assuming that **groupB** is the group that you want to attack, images watermarked by that group downloaded from the website will be named

`groupB_imageName.bmp`

Assuming that **groupB** is the group that you want to attack and you are **groupA**, images attacked by your group must be named

`groupA_groupB_imageName.bmp`

Scoring Information

Scores will be assigned to each group's performance during the competition according to the following tables:

- **EMBEDDING QUALITY**: aim at a higher quality of the watermarked image for more points.

WPSNR	POINTS
$35 \leq \text{WPSNR} < 50$	1
$50 \leq \text{WPSNR} < 54$	2
$54 \leq \text{WPSNR} < 58$	3
$58 \leq \text{WPSNR} < 62$	4
$62 \leq \text{WPSNR} < 66$	5
$\text{WPSNR} \geq 66$	6

- **ROBUSTNESS**: average WPSNR of the group's watermarked images successfully attacked by other groups

WPSNR	POINTS
$35 \leq \text{WPSNR} < 38$	6
$38 \leq \text{WPSNR} < 41$	5
$41 \leq \text{WPSNR} < 44$	4
$44 < \text{WPSNR} < 47$	3
$47 \leq \text{WPSNR} < 50$	2
$50 \leq \text{WPSNR} < 53$	1
$\text{WPSNR} \geq 53$	0

- **ACTIVITY:** percentage of groups attacked

% OF GROUPS ATTACKED	POINTS
> 30%	2
> 60%	4
> 90%	6

- **QUALITY:** number of images that you attacked obtaining a WPSNR higher than the challenge's average (AVG) WPSNR.

ATTACKED IMAGES WITH WPSNR > AVG WPSNR	POINTS
1-5	1
6-10	2
11-15	3
> 15	4

- **BONUS:** you will be awarded 2 extra points
 - if you successfully attacked a group that no one else attacked
 - if you were not attacked by anyone (assuming that your code is working)

These bonuses will be removed during the verification phase if they are caused by errors in the code or images.

Penalties

A 2-point penalty in the competition results will be applied for each transgression of the discussed rules, including the following.

- You miss a deadline.
- You have to change one of the files uploaded to the website after the conclusion of the defense phase.
- The detection function successfully finds a watermark in non-watermarked images, in several unrelated images (e.g., images watermarked by other groups), or destroyed images ($\text{WPSNR} \leq 25\text{db}$)
- Your detection code opens pop-up windows or prints on the screen.
- Your detection code takes more than 5 seconds to run.