Watermarking Lab — Notes with Inline Clarifications

# Lab constraints

The attacks that can be used can be only the ones listed in the lab.  
How we parametrize them is up to us.  
We cannot use the original image, only the watermarked one.   
No blind watermarking techniques are allowed; we must implement non-blind techniques.

✅ Clarification: Your notes indicate that blind watermarking is \*not allowed\*. Make sure the detection/extraction method uses some form of side information (like the original watermark or a key).

# Least Significant Bit (LSB)

Least Significant Bit (LSB) -> is a blind technique.  
When substituting LSB, the visual difference won't be visible.  
It is imperceptible, but there is always a trade-off: even if not visible, it is not robust.  
It is the easiest technique: pixels become bits and it replaces the least important bit.  
It is very invisible; it can be made less invisible but more robust by using a different bit than the absolute least significant one.  
  
Works:  
read the image  
Those three lines from N onwards can be used to create the watermark.  
We can decide how we want to embed the watermark once or multiple times.  
**IN THE PROVIDED CODE: Type of the pixel = string, not Integer.**  
After that it must be encoded back into pixels.

LSB is a blind, fragile method; works conceptually, but not robust against compression or noise.

# PSNR

Is a measure of the differences between the original image and the watermarked image.  
The problem is it offers something perceptible to measure, but without taking into account how the human system works.

**The human eye is most sensitive to green colors.**  
It is also more sensitive to flat (smooth) regions. Distortions are seen more easily on flat areas than on textured ones.  
  
When implementing the embed code we must check how PSNR behaves with robustness; we must find a balance.

🔍 Clarification: Correct observations — PSNR doesn’t reflect the human visual system. Use both PSNR and WPSNR for better evaluation, balancing visibility and robustness.

# WPSNR

Weighted PSNR (WPSNR) considers human visual sensitivity. Keep **WPSNR > 35 dB** to maintain acceptable quality.

ℹ️ Clarification: Weighted PSNR (WPSNR) adds perceptual weighting and correlates better with human vision. Aim for WPSNR ≥ 35 dB to maintain acceptable image quality.

# LSB Detection

It is an example how watermarking techniques are weak.  
First of all we are **not allowed to use blind techniques** but **non-blind**.  
How we measure if the watermark is similar to the one we embedded = Similarity between the Extracted Watermark and the Original Watermark.  
We will have our image with embedded watermark and also the attacked watermarked image; we will compare the similarities.  
  
***Attention to Det Tec i-a Iw not direct W = Use the watermarked image (Iw) in detection, not watermark W directly*:**  
So we must ensure that the watermark for **detection and attack is extracted from an image** **that has a watermark**; if it does not, it is not possible.

✅ Clarification: Correct — detection must only be applied to images that contain a watermark. For evaluation, compare extracted vs original watermark using a \*\*similarity metric\*\* like correlation.

# Spread Spectrum Watermarking

We **must not replicate the VS Code code 1:1**, but **implement more robust things** (using the notebook where RGB domains are discussed, we should add those).   
Also prof. Bato will send us a paper, it would be good to implement those as well.  
  
**Spread Spectrum Watermarking**  
This is good because it **is non-blind**.  
If we add bit information in **large frequency** it is not perceptible.  
  
The **trade-off**: it depends on the coefficient we use.

It would be good to be in medium frequency, or HL, LH, but not low-low.  
  
**High-mid frequencies should be targeted.**  
  
Additive = more **imperceptible**, more invisible, but **less robust**.  
Multiplicative = more **visible**, but **more robust** -> **alpha value must be considered**, because if it is **too large it destroys the image** (values should be **up to 0.2**).

⚙️ Clarification: All correct — target mid/high-mid frequencies (HL/LH). ✅ \*\*Note:\*\* Keep α ≤ 0.2, but test empirically; multiplicative gives robustness, additive gives invisibility.

# Receiver Operating Characteristic (ROC) Curve

**How to set the threshold using the Receiver Operating Characteristic (ROC) Curve**  
It is used for **binary classification**, like the one used in Detection, when we decide whether the watermark is present or not.  
**We cannot use an FP threshold — we can choose between 0 and 0.1, but it is up to us what we choose.**  
If it is robust we can say in the presentation that we set it to 0; if not we can use 0.1.  
It may not have a practical advantage, but it helps to show the dedication we put into the code.  
**We get points at presentation if we motivate well the decision to use 0 or 0.1.**  
  
Attack parameters presented in the previous lab must be randomized and the execution order also randomized to see how the ROC behaves.  
  
Before using ROC code we must play with the attacks.  
WPSNR < 35 = 0 points, because we destroy the image.  
**All attacks can remove the watermark, but they must preserve the image.**  
  
60% of protective attacks when we implement defense is sufficient and then we can focus on attacks.  
  
Do not embed the watermark once; redundancy is important.

✅ Clarification: ROC curve use is correct. Randomizing attacks and parameters is good for fair evaluation. WPSNR < 35 → unacceptable image quality; redundancy improves robustness. ✅ Keep the 60% rule as an internal success target.

# Summary and Reminders –– ChatGPT Generated

## 1. General Rules

* Use **only the attacks listed** in the lab.
* The **parameterization** and randomization of attacks are up to us.
* The **original image** cannot be used; only the provided **watermarked image**.
* **Blind watermarking is not allowed** – we must implement **non-blind techniques**.

## 2. LSB (Least Significant Bit)

* LSB is a **blind and fragile** technique, visually imperceptible but not robust to compression or noise.
* In the provided code, pixels must be treated as strings.
* Using a higher bit plane (e.g., 2nd LSB, not the LSB) slightly increases robustness but decreases quality.
* LSB can be used as a **baseline example**, not as the main robust method.

## 3. Image Quality Metrics

* **PSNR** measures pixel-wise distortion but does not reflect human perception.
* **WPSNR** is more perceptual; it accounts for sensitivity to color and texture.
* Maintain **WPSNR > 35 dB** to ensure acceptable visual quality.
* When embedding, aim for a balance between imperceptibility and robustness.

## 4. Detection

* Detection must be applied only to **images that contain a watermark (Iw)**, not directly to the watermark (W).
* Evaluate detection by measuring **similarity or correlation** between the extracted watermark and the original one.
* Include tests on both **watermarked** and **unwatermarked** images to validate low false positives.

## 5. Spread Spectrum Watermarking

* Do not replicate the VS Code example directly; implement **more robust variations** using the material discussed in the RGB-domain notebook or the paper provided by prof. Bato.
* Target **medium or high–mid frequencies** (e.g., HL, LH subbands); avoid very low frequencies.
* **Additive** embedding: less visible, less robust.
* **Multiplicative** embedding: more robust but may degrade image quality.
* Keep **α ≤ 0.2**, adjusted empirically through testing.

## 6. ROC Curve and Attack Evaluation

* The **ROC (Receiver Operating Characteristic)** curve is used to decide the detection threshold (presence or absence of watermark).
* The false positive rate (**FPR**) can be set between **0 and 0.1**, depending on the robustness achieved.
  + If highly robust → FPR ≈ 0
  + If moderately robust → FPR ≈ 0.1
* **Randomize attack parameters and order** to obtain reliable performance measurements.
* If **WPSNR < 35 dB**, the image is considered destroyed and not valid for scoring.
* The watermark should **survive approximately 60% of attacks** while preserving image quality.
* Do not embed the watermark only once — **redundancy is required** for stability.

## 7. Implementation Checklist

| **Criterion** | **Requirement** |
| --- | --- |
| Method | Non-blind watermarking |
| Image Quality | WPSNR > 35 dB |
| Robustness | Survives ≥ 60% of attacks |
| Frequency Domain | Medium / high–mid coefficients used |
| Embedding Strength | α ≤ 0.2 (tested empirically) |
| Detection | Correlation or similarity-based |
| Evaluation | ROC curve generated with randomized attacks |
| Redundancy | Watermark embedded multiple times |