# A Digital Signal Processing Report on
# Image Steganography using LSB Matching

**By: Himanshu Sharma**
**Roll Number:** 1610110149
**Dept. of Electrical Engineering (ECE)**

**Shiv Nadar University,**
**Gautam Buddh Nagar, Greater Noida, Uttar Pradesh 201314**

Under the guidance of Professor Vijay K. Chakka

# 1 Paper Comprehension

*Image Steganography* is the art of hiding information inside a digital image. Steagnography does not restrict to only plain text but it includes text, video, audio and image hiding also. In this report, the author has restricted himself to text data type only. Even when doing text embedding, there are immense possible algorithms to choose from. The most popular among them is the LSB replacement or Least Significant Bit replacement. With this type of algorithm, the least significant bit of each pixel (only one channel out of the RGB pallete) is changed by a bit decided according to the message bit. Therefore, we should not expect much change in the image after data hiding because the least significant bit is changed. For example, lets take a blue channel with value of 145. In binary, it is equivalent of 10010001. If by some technique, the last bit is changed to 0, then the decimal equivalent would become 144, which does not change the image much. This is the power of LSB replacement. It basically relies on the fact that hiding message bits in the LSB of each pixel does not affect the image much, in fact, the image practically remains as it is. There are, however, techniques now in modern signal processing science which can detect that some data is hidden in the image or not, a field called *steganalysis.* A successful hiding algorithm would be that, that would allow high payload and still look similar to the original image.

## 1.1 LSB Matching

In LSB matching, if the message bit does not match with the pixel's LSB, then ±1 is randomly added to that pixel value. Unlike LSB replacement, where the pixel's LSB is just replaced by the message bit, here, a set of conditions are used to modify the pixel. LSB Matching could not be detected using the techniques used to detect LSB replacement. However, now it has been proved that LSB matching acts like a low pass filter on the digital images and therefore, this fact is utilized to detect whether LSB matching is applied on an image or not.

Usually, two consecutive pixels are taken alongwith two consecutive message bits. The consecutive pairs are chosen randomly based on the *pseudo-random number generator* (PRNG). If the LSB matching is used as it is, then any pixel could be chosen with every pixel pair having equal chances of being selected by the algorithm. This has a flaw. This type of approach makes it difficult to disguise a changed pixel to the pixel surrounding it. For example, if lot of pixels are changed in a close vicinity, then they could be easily identified if the region is light in color, like sky which is light blue in color. The paper chosen here tries to rectify this problem by chosing those pixels on the image which lie on the edges. Edges are usually sharper than the surrounding regions and therefore its not easy to identify the change in pixel color on the edges. Similar papers have already been published. All of them suggest to use what is called the *pixel-value difference* (PVD).

In PVD, what we do is that when we try to hide a message bit in a pixel's LSB, the pixel value is compared with its neighbouring pixels. If the difference is large, then more bits can be accomodated in that region without them being easily identified. Why? Because if there is a large difference in the pixel values then on changing the pixel value will not generate any significant difference. For example, if a pixel that is to be modified has a value of 45 and it's neighbouring pixel has a value of 145, then the difference $\Delta = 145 - 45 = 100$. Now, if by some technique if this pixel value if changed to 46, then the difference would become $\Delta' = 145 - 46 = 99$. To a human eye, this difference is not accountable. PVD is a good approach, in fact, far more better that PRNG because it utilizes the fact that sharp changes can be used to hide the information.

In both LSB replacement and LSB matching, a travelling order is generated using PRNG which also acts as a key for decoding the stego-image. In both of these algorithms, the LSB of the selected pixel becomes equal to the message bit. According to the algorithm, the if the two consecutive pixels are $x_i$ and $x_{i+1}$ and the consecutive message bits are $m_i$ and $m_{i+1}$, then the pixels are modified such that $x_i$ becomes $x_i^{'}$ and $x_{i+1}$ becomes $x_{i+1}^{'}$ and the following relation holds.

$$LSB(x_i^{'}) = m_i \text{ and } LSB\Big(\Big\lfloor \frac{x_i^{'}}{2} \Big\rfloor + x_{i+1}^{'}\Big) = m_{i+1}$$

From experiments done by the authors of the paper it has been shown that even if the cover image has rough textures, it will still have smooth regions in every $5 \times 5$ non-overlapping blocks. So, if by any chance, a pixel is selected in that region for message hiding then it could be easily identified. This is what the paper aims to solve.