**III.1 Introduction**

Deep Neural Networks (DNNs) are increasingly deployed and commercialized in a wide variety of real-world scenarios due to the unprecedented performance they achieve. Training a DNN is a very expensive process that requires: (i) the availability of massive amounts of often proprietary data capturing different scenarios within the target application; ii) extensive computational resources; iii) the assistance of Deep Learning (DL) experts to carefully fine-tune the network topology (e.g., the type and number of hidden layers), and correctly set the training hyper-parameters, like the learning rate, the batch size, etc. As a consequence, high performance DNNs should be considered as the intellectual property (IP) of the model owner and be protected accordingly. Inspired by the use of classical watermarking techniques for the protection of property rights associated to multimedia contents, DNN watermarking is receiving increasing attention, and several works have been published leveraging on digital watermarking to address IP protection in the DL domain.

In the last two decades, watermarking technology has been applied to protect multimedia documents, and a vast body of literature has been developed as summarized in several books and surveys. The watermark can be injected into the host document by adding to it a low amplitude, often pseudo-random, signal, either directly in the sample domain or in a properly transformed domain. In the case of still images, for instance, the watermark can be embedded in the spatial domain by adding a low amplitude spread spectrum signal or by substituting the least significant bits (LSB) of the pixel values. In the case of transformed domain watermarking, several reversible transforms like the Discrete Cosine Transform (DCT), the Discrete Wavelet Transform (DWT), or the Discrete Fourier Transform (DFT) are widely used to embed the watermark in a robust and imperceptible way [1].

With all that been said, in this chapter we’re going to take a deep dive with extreme details on how watermarks are embedded/extracted in DNN with its taxonomy and all the requirements needed for the process to achieve decent results like setting up or generating a good dataset for the training, testing the efficiency of the algorithm by applying a different verity of attacks (compression, sharpening, gaussian blur ... etc.).

Also getting familiar with some terminology in DNN and image processing terms in general to keep track and get a full understanding of this survey.

**III.2. Why DNN watermarking:**

We deal with images in many domains. Now we are processing signals (images) in frequency domain. Since this Fourier series and frequency domain is purely mathematics, till now, all the domains in which we have analyzed a signal, we analyze it with respect to time. But in frequency domain we don’t analyze signal with respect to time, but with respect of frequency.

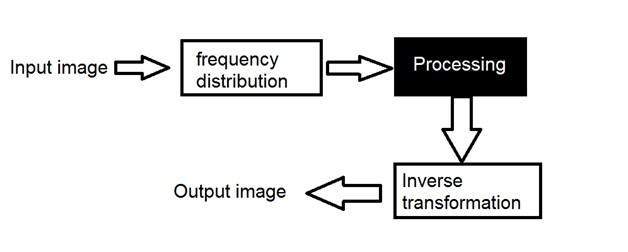
In spatial domain, we deal with images as it is. The value of the pixels of the image change with respect to scene. Whereas in frequency domain, we deal with the rate at which the pixel values are changing in spatial domain [2].

For simplicity, Let’s put it this way.



**Figure III.1:** Special domain overall model.

In simple spatial domain, we directly deal with the image matrix. Whereas in frequency domain, we deal an image like this: We first transform the image to its frequency distribution. Then our black box system performs whatever processing it has to performed, and the output of the black box in this case is not an image, but a transformation. After performing inverse transformation, it is converted into an image which is then viewed in spatial domain. It can be pictorially viewed as:



**Figure III.2:** Frequency domain overall model.

A signal can be converted from time domain into frequency domain using mathematical operators called transforms. There are many kinds of transformation that does this we discussed some of them in the previous chapter.

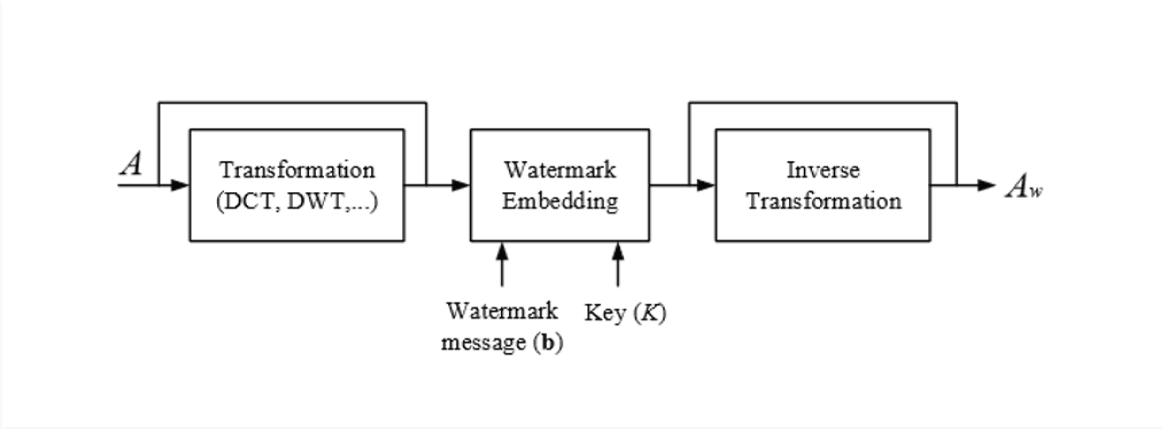
As an answer to the question as to why we use DNN in watermarking and not a classic watermarking, first of all because the classic interpretation of images does not work if a marked image gets modified, for instance LSB method wouldn’t be able to extract the right watermark if the cover image gets modified by a compression algorithm although the original and the compressed image does not seem to be different, but a DNN would even if some pixels values changed on the marked image. Second of all someone might ask, well why we use a deep neural network not just a network, well because Deep Learning and Neural Networks are so closely related, it’s difficult to tell them apart on the surface. However, you’ve probably figured out that Deep Learning and Neural Networks are not exactly the same thing. Deep Learning is associated with the transformation and extraction of features that attempt to establish a relationship between stimuli and associated neural responses present in the brain, whereas Neural Networks use neurons to transmit data in the form of input to get output with the help of the various connections.

**III.3. A taxonomy of DNN watermarking:**

In this section, we present a taxonomy of DNN watermarking techniques. The taxonomy takes into account the similarities and dissimilarities between multimedia and DNN watermarking, some of which have already been outlined in the introduction. To start with, we review the most important classification criteria used to categorize classic watermarking algorithms, discussing their applicability to the DNN case. Later, we introduce some unique classification criteria based on the peculiar characteristics of DNN watermarking. Eventually, we discuss the relationship between the various watermarking classes.

**3.1. Classical watermarking models**

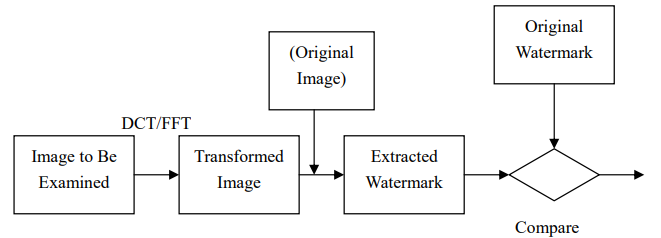
Classically, digital watermarking aims at embedding a certain message, called watermark, within a hosting multimedia content, often, but not only, for copyright protection. Watermark injection follows the general model depicted in Figure 3. The to-be-watermarked content *A* is possibly transformed so that the watermark is embedded in a different domain (e.g., the DCT or wavelet domain for still images), then the watermark is injected within the document,  
with the possible intervention of a watermarking key *K*. The watermarked content is then brought back into the original domain to obtained the watermarked document *Aw*. As simple as this scheme may seem, it contains some implicit assumptions that are not necessarily satisfied in the case of DNN watermarking: i) the existence of a to-be-watermarked content *A* and *ii*) the consequent possibility of measuring the distortion between the original and the watermarked contents, *A* and *Aw*. This is not necessarily the case with DNN watermarking, where the network weights, which are going to host the watermark, may not exist perse, since they are generated contextually to watermark embedding during the training phase. Moreover, the distortion introduced by the watermark cannot be evaluated directly by measuring the difference between *A* and *Aw*. Rather, the impact of the watermark must be measured by evaluating the performance achieved by the watermarked model. Note that watermark embedding (and later on watermark recovery) may require the knowledge of a secret key *K*. Such a key, whose main goal is to introduce some secrecy within the watermarking system, is commonly used to parameterize the embedding process and make the recovery of the watermark impossible for non-authorized users who do not know *K*.



**Figure III.3:** Overall view of classical watermark embedding.

Most frequency-domain algorithms make use of the spread spectrum communication technique. By using a bandwidth larger than required to transmit the signal, we can keep the SNR at each frequency band small enough, even the total power transmitted is very large. When information on several bands is lost, the transmitted signal can still be recovered by the rest ones. The spread spectrum watermarking schemes are the use of spread spectrum communication in digital watermarking. Similar to that in communication, spread spectrum watermarking schemes embed watermarks in the whole host image. The watermark is distributed among the whole frequency band. To destroy the watermark, one has to add noise with sufficiently large amplitude, which will heavily degrade the quality of watermarked image and be considered as an unsuccessful attack [3].

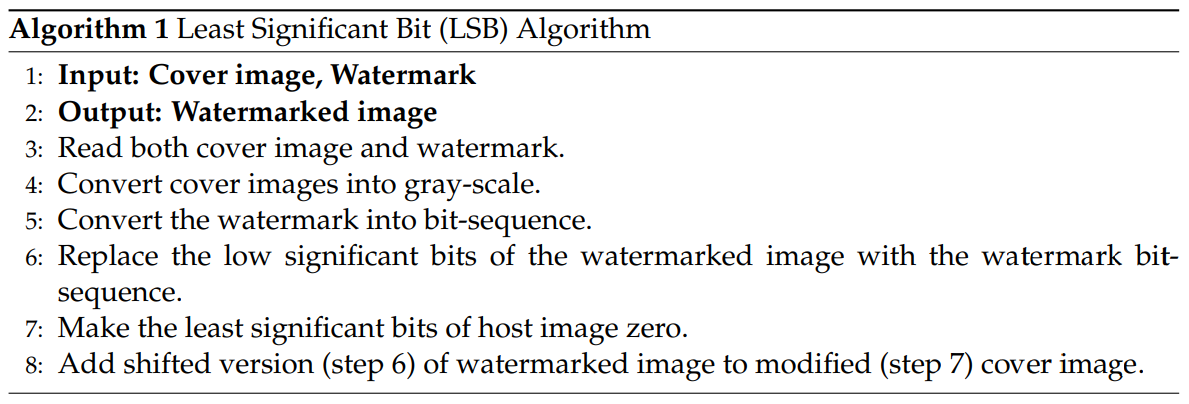
Figure 4 illustrate the watermark embedding and detection /extraction in frequency domain, respectively.



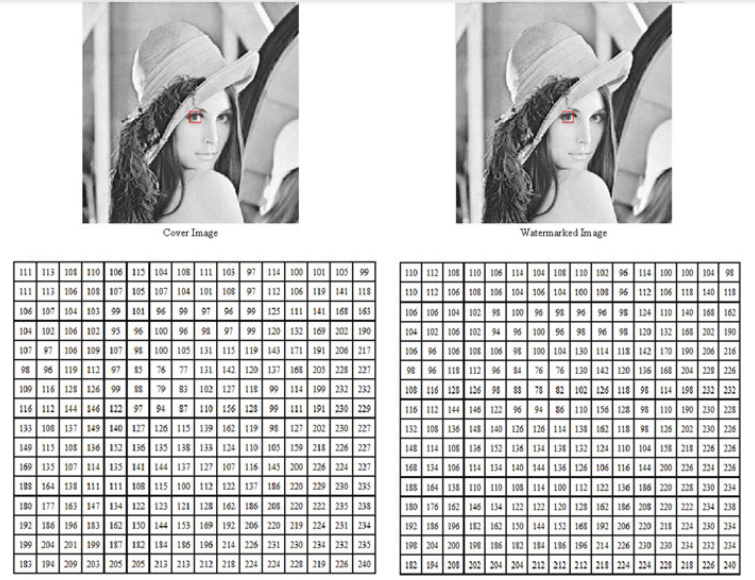
**Figure III.4:** Watermark detection/extraction model in frequency domain

One major reason why frequency domain watermarking schemes are attractive is their compatibility with existing image compression standards, in particular, the JPEG standard. The compatibility ensures those schemes a good performance when the watermarked image is subject to lossy compression, which is one of the most common image processing methods today. In consequence, those schemes become particularly useful in practical applications on the Internet.

We’ve seen one of the classical methods in frequency domain, now let us take a look at one of the common classical methods in special domain called LSB which is the simplest and the most commonly used technique in spatial domain, for a given image each pixel is represented by an 8-bit sequence, the last bits (least significant bit LSB) of the pixel values of that image contain less relevant information while the most information is contained in the most significant bits (MSBs). Thus, the modification of LSB bits does not cause perceptible changes. Before embedding the watermark, it is first encrypted then some random pixels of the cover image are selected using a key, which determines the pixels that will be modified by the embedding process. This technique of watermarking consists of replacing the LSB of the pixel values in the carrier image with the watermark’s values. Thus, the watermark can be hidden partially or completely. To extract the watermark, embedding algorithm is reversed. Figure 5 illustrates the LSB method sample of watermark embedding (to extract the watermark, same steps are applied in the inverse order.), moreover, embedding algorithm steps are shown in Algorithm 1 [4].



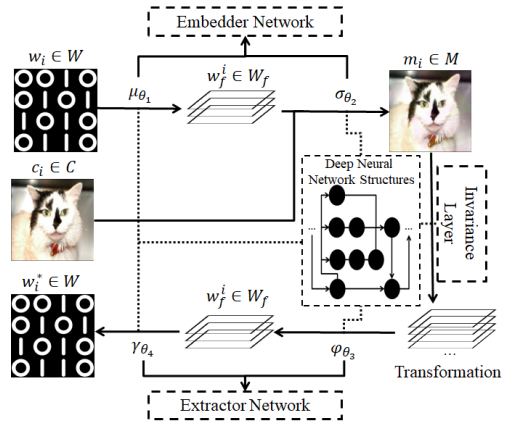
Besides the easy implementation of this method (See Algorithm 1), it provides high perceptual transparency, and resistant against cropping attacks. Even though, LSB techniques are less robust to common signal processing operations and sensitive to noise and compression which can simply destroy the watermark.



**Figure III.5:** LSB watermarking sample of embedding and extraction of the watermark.

**3.2. Modern watermarking models:**

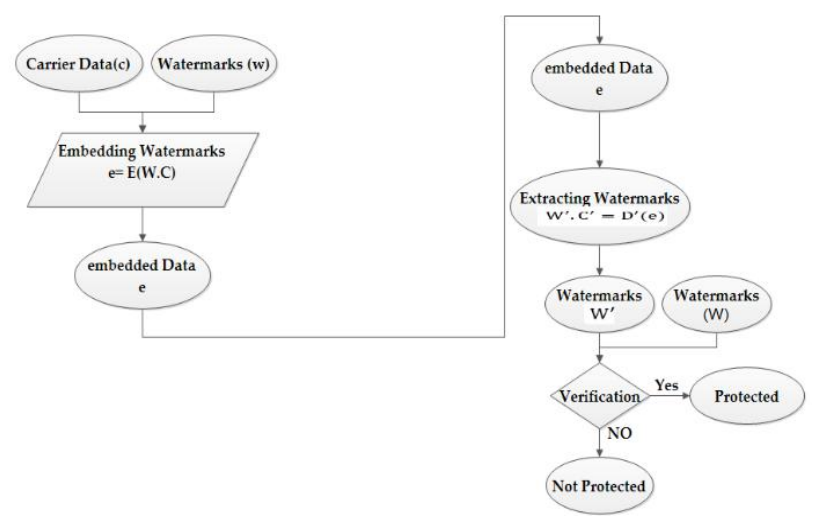
Xin Zhong and Frank Y. Shih developed an automated image watermarking system using deep learning networks based on three main motivations. First, exploring the fitting ability of deep learning models in learning the rules of watermark embedding is helpful in developing an automated system. Second, the proposed system is tested on the application of watermark extraction from camera resamples, providing a potential solution towards this challenging issue. Third, image watermarking is viewed from a novel perspective – an image fusion task between the cover-image and the latent spaces of the watermark, where the fused result (i.e., the marked-image) contains the watermark while references the visual appearance of the cover-image. Were they proposed the following model:



**Figure III.6:** The architecture of DNN watermarking system

They apply deep neural networks 𝜇𝜃1, 𝜎𝜃2, 𝜑𝜃3 and 𝛾𝜃4 with parameters 𝜃1, 𝜃2, 𝜃3 and 𝜃4 to learn the mapping functions 𝜇, 𝜎, 𝜑 and 𝛾. The architecture of the proposed image watermarking system is shown in Fig. 2, where 𝑤𝑖, 𝑤𝑓 𝑖, 𝑐𝑖, and 𝑚𝑖 are the examples of the spaces 𝑊, 𝑊𝑓, 𝐶 and 𝑀. 𝜇𝜃1 and 𝜎𝜃2 are named as the embedder network, and 𝜑𝜃3 and 𝛾𝜃4 are named as the extractor network [5].

Were other researchers like Farah Deeba and She Kun propose slightly different models as shown below in figure 7:

****

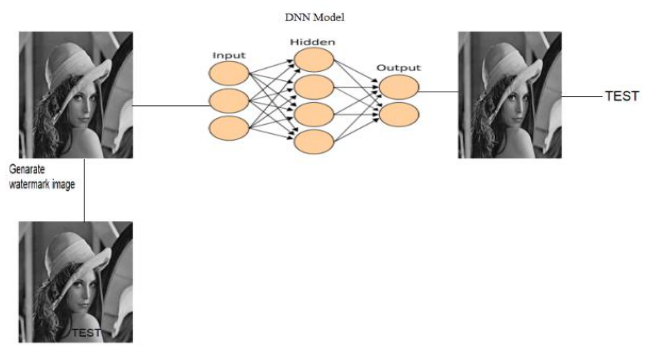
**Figure III.7:** Watermarking process model.

From Figure 7 we can recognize the complete typical watermarking process. An embedding algorithm "E" embeds the pre-trained watermarks W into the carrier data C that should be protected from attacks. When embedding is done the data stored in (e=E (W, C)) where decryption is applied to extract the watermark "W" from e [6].

**3.3. DNN watermarking models:**

Owing to the many neurons in DNN models, there is significant redundancy in the network; making a network deeper is a promising way to improve the performance. It is reasonable to prune such redundant neurons to reduce the model size as well as to reduce computational costs. It is an NP-hard problem to find the best combination of neurons to be pruned, from among the millions of parameters in a DNN model. Some heuristic pruning methods have been developed to identify relatively less important components in DNN models and retrain the pruned model to recover the model’s performance.

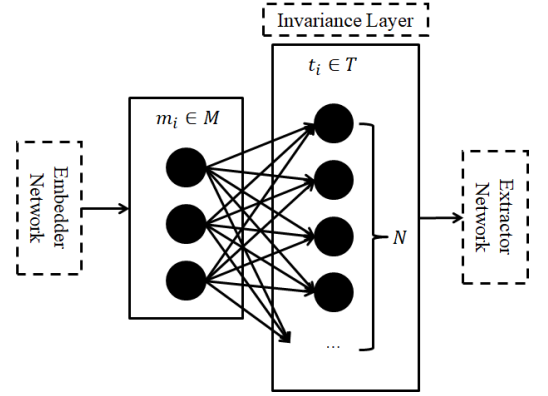
Thus, to create a robust DNN watermark, it is necessary to consider the effects of pruning as well as the changes during retraining.



**Figure III.8:** Workflow of DNN watermarking.

The above Figure 8 shows the DNN watermark Framework. When generating Watermarks these watermarks shown as the fingerprint in ownership as well. It first creates the watermark; these watermarks reported as the fingerprint verification. Then these watermarks embed to DNNs through training. The DNN automatically learns the patterns of watermarks and retains them. And finally for verification after embedding our models is strong enough to verify the ownership. Following newly generated models able to verify ownership so that the owner can prove by simple sending watermark as input and checking through output by deploying DNN model.

Xin Zhong and Frank Y. Shih in [5] proposed an invariance layer introduces a function 𝜏: 𝑀 → 𝑇 that maps space 𝑀 to an overcomplete transformation space 𝑇. The neurons in this layer are activated in a sparse manner not only to allow a possible loss in 𝑀 for robustness but also to enhance computational efficiency. As shown in Figure 9, it converts a 3-channel instance 𝑚𝑖 of M To tolerate the distortions on the marked-images without considering all possible attacks.



**Figure III.9:** The invariance layer.

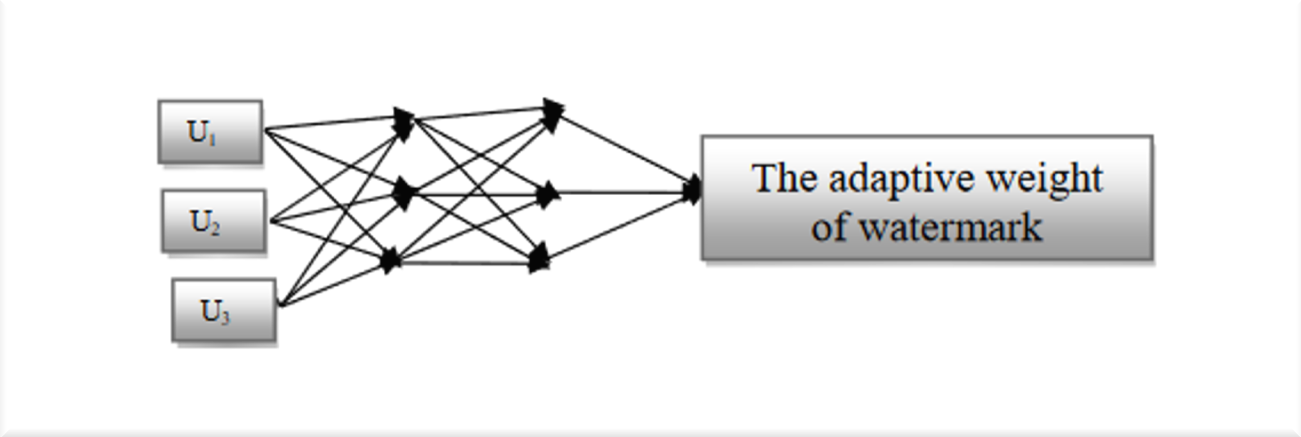
**4.5. Other Watermark Requirements**

Based on the choice of a specific watermarking modality among those discussed before, and on the envisaged application, other criteria may need to be satisfied.

**Generality:** To be effective, a DNN watermarking algorithm should be applicable to a wide variety of architectures carrying out different tasks. In this sense, the generality of an algorithm refers to its applicability to architectures and models other than those on which the algorithm was initially tested and developed. This also implies that the performance of the algorithm does not depend too heavily on the to-be-marked model.

**Efficiency:** Efficiency refers to the computational overhead needed to train the DNN by simultaneously teaching the model to carry out the target task and embedding the watermark. Training a DNN is always an expensive process, embedding the watermark into the DNN should not add an unaffordable extra burden to it.

**III.5. Application of Deep Neural Network in Digital Image Watermarking:** Artificial neural network is important embranchment of artificial intelligence field, and it has certain advantage on the aspect of simulating biology neural computation [44], [45]. It has self-learning, self-organizing, association of ideas, blur extend abilities, etc. and has great comparability with human visual system. Therefore, the selected NN in this paper is a feed-forward NN. There are three layers including an input layer, hidden layers and an output layer. The desired outputs of NN are the maximum watermarking strength. Figure 7 shows   
the architecture of this network. The first layer is to present the input variables of network. Each HVS feature vector has the elements to be input variables in this study. The four feature components form a feature vector U = (U1, U2, U3, U4). The U1, U2, U3 and U4 represent the sensitivity value of luminance, frequency, texture, and entropy of respectively. Each u represents one of the input components in NNS algorithm and each pixel generates a corresponding feature vector u. These inputs are then fed to a hidden layer using tan sigmoid   
activation functions (output in the range -1 to 1); this is then fed to the output layer. During training samples, information is propagated back through the network and used to update connection weights. It repeats learning many times for every example in the training set until it has minimized the output errors. This neural network, when combined with the available rules in perceptual watermarking of an image and ability to query the network, does provide an automatic system for generating near maximum power invisible watermarked images.   
This system does not fully replace humans, its they are still needed to generate the training data and should check the results periodically to ensure the neural network is working at peak performance.

****

**Figure III.7:** Architecture of NN with textural features.

**III.6. conclusion**

The demand of methods for protecting the Intellectual Property Rights associated to DNNs has pushed researchers to develop a new class of algorithms to embed a watermark into DNN models. In a few years, several techniques have bene proposed, sometimes based on naive arguments and sometimes by relying on solid theoretical bases. By looking at the methods proposed so far, it is evident that watermarking a DNN model presents some peculiarities that must be taken into account when trying to apply general multimedia watermarking principles to DNNs. For this reason, we have started our review by highlighting the main differences and similarities with classical multimedia watermarking, and introducing a new specific taxonomy explicitly thought to highlight the characteristics of the various DNN watermarking algorithms proposed so far. Then we have reviewed the main algorithms available for each watermarking category, pointing out the main advantages and drawbacks characterising the various approaches. By the light of the properties and limits of the watermarking algorithms proposed so far, the most important challenges researchers are going to face with in the years.

**References :**

[1]: A survey of deep neural network watermarking techniques by Yue Lia, Hongxia Wangb and Mauro Barnic.

[2]: <https://www.tutorialspoint.com/dip/introduction_to_frequency_domain.htm>

[3]: Research on Image Watermarking Algorithm Based on DCT By Z.J.Xu and Z.Z. Wang.

[4]: A Robust Watermarking Approach for Images Authentication and Traceability By Wassila BELFERDI page 19.

[5]: A Robust Image Watermarking System Based on Deep Neural Networks by Xin Zhong and Frank Y. Shih.

[6]: Digital Watermarking Using Deep Neural Network by Farah Deeba and others.