# LSSD: a Controlled Large JPEG Image Database for Deep-Learning-based Steganalysis "into the Wild"

Hugo Ruiz, Mehdi Yedroudj, Marc Chaumont, Frédéric Comby, Gérard Subsol

*ICPR'2021, International Conference on Pattern Recognition, MMForWILD'2021, Workshop on MultiMedia FORensics in the WILD, Lecture Notes in Computer Science, LNCS, Springer. January 10-15, 2021, Virtual Conference due to Covid (formerly Milan, Italy). Version of December 2020.*

# LSSD: a Controlled Large JPEG Image Database for Deep-Learning-based Steganalysis "into the Wild"

Hugo Ruiz[1 [0000−0002−2806−2428]], Mehdi Yedroudj[1 [0000−0001−6404−3876]], Marc Chaumont[1,2 [0000−0002−4095−4410]], Frédéric Comby[1 [0000−0001−7157−4296]], and Gérard Subsol[1 [0000−0002−7461−4932]]

[1] Research-Team ICAR, LIRMM, Univ. Montpellier, CNRS, France
{hugo.ruiz, mehdi.yedroudj, marc.chaumont, frederic.comby, gerard.subsol}@lirmm.fr
[2] University of Nîmes, France

**Abstract.** For many years, the image databases used in steganalysis have been relatively small, i.e. about ten thousand images. This limits the diversity of images and thus prevents large-scale analysis of steganalysis algorithms.
In this paper, we describe a large JPEG database composed of 2 million colour and grey-scale images. This database, named LSSD for Large Scale Steganalysis Database, was obtained thanks to the intensive use of "controlled" development procedures. LSSD has been made publicly available, and we aspire it could be used by the steganalysis community for large-scale experiments.
We introduce the pipeline used for building various image database versions. We detail the general methodology that can be used to redevelop the entire database and increase even more the diversity. We also discuss computational cost and storage cost in order to develop images.

**Keywords:** Steganalysis · scalability · million images · "controlled" development · mismatch

## 1 Introduction

Steganography is the art of hiding information in an non suspicious medium so that the very existence of the hidden information is statistically undetectable from unaware individuals. Conversely, steganalysis is the art of detecting the presence of hidden data in such supports [8]. JPEG images are attractive supports since they are massively used in cameras and mobile phones and in all media of communication on the Internet and social networks. In this paper, we will then focus on steganography and steganalysis in JPEG images.

In 2015, steganalysis using Deep Learning techniques emerged [2], and nowadays, they are considered as the most efficient way to detect stego images (i.e. images which contain a hidden message). Moreover, GPU computation capabilities increase regularly, which ensures faster computing speed which reduces

learning time. So, performances of steganalysis methods based on Deep Learning have significantly improved.

Nevertheless, in most cases, the performance of steganalysis based on Deep Learning depends on the size of the learning image database. To a certain extent, the larger the database is, the better the results are [16] [15]. Thus, increasing the size of the learning database generally improves performance while increasing the diversity of the examples.

Currently, the most significant database used in steganalysis by Deep Learning is made of one million JPEG images [20] excerpted from the ImageNet database, which contains more than 14 million images.

That said, databases created in "controlled" conditions, that is to say, with the full knowledge of the creation process of the images so that the development is repeatable, are not very big in comparison. Indeed, we can mention, as a "controlled" database the BOSS database [1] with a size of only 10,000 images and the Alaska #2 database, with a size of 80,000 images [5].

In this paper, we present the "controlled" *Large Scale Steganalysis Database* (LSSD) which is a public JPEG image database, made of 2 million images, with a colour version and a grey-scale version, and which was created for the research community working on steganalysis.

One important aspect when creating an image database for steganalysis is to have diversity to get closer to reality [12]. This "diversity" mainly depends on the ISO and the "development" process of the RAW image that is captured by the camera sensor. ISO is a measure of the sensitivity to light of the image sensor and if available, can be notified in the metadata associated with the JPEG image, i.e. in the EXIF metadata.

As in analogical photography, the "development" process consists in applying image processing operations (demosaicing, gamma correction, blur, colour balance, compression) in order to transform the RAW image into a viewable image in a standard format. The RAW image, when the camera is made of a colour filter array (CFA) of type Bayer filter (which is majority the case), is a unique 2D matrix containing 50% green, 25% red and 25% blue. In order to "control" this diversity, it is possible to tune the different development parameters, without modifying the ISO parameters, and get different developments and then different JPEG versions of the same RAW image. Thus, if we want to increase the size of the database to more than 10 million images, it is necessary to implement a well-thought-out procedure to automate the "controlled" generation, to optimize the processing time as well as the storage volume.

By controlling the development, it becomes possible to get large databases which can be used for the learning or the test phases of Deep-Learning based steganalysis algorithms. We can then conduct an objective and repeatable evaluation of the performances of these algorithms. In particular, it will allow researchers to work on one of the major challenges in the steganalysis field: the "Cover-Source Mismatch". Cover-Source Mismatch (CSM) is a phenomenon that occurs when the training set and the test set come from two different sources,

causing bias in the Deep-Learning learning phase and resulting in bad results in the test phase.

In Section 2, we detail the whole development procedure that is used for the generation of the LSSD database. In Section 3, we explain how to use the LSSD Database to create a learning set and a test set for Deep-Learning steganalysis applications. We also emphasize the problem of computational and storage cost for the creation of those sets.

## 2 A "controlled" procedure to get a JPEG image

### 2.1 RAW image sources

To build a consistent "controlled" base, we chose to gather a maximum of RAW images, i.e. which are composed of the original sensor data. More precisely, we use the file that contains the RAW data of the sensor before any lossy compression (JPEG for example), and before any transformation required for its visualization on screen. It is important to note that each manufacturer adapts the data format to its hardware and then that we can find many formats (e.g., .dng, .cr2, .nef. . . ).

At the contrary of JPEG images, RAW images are extremely rare on the Internet because they are large files and used by very few people. The size of a RAW image is usually around $3,000 \times 5,000$ pixels. Since data are in "raw" format, it represents a lot of information to store. It is therefore rare that Web sites, even those specialized in photography, dedicate a specific storage space to this kind of images.

Table 1: Number of images and devices used in each database.

| **Database Name** | **number of images** | **number of devices** |
|---|---|---|
| ALASKA2 [3] | 80,005 | 40 |
| BOSS [4] | 10,000 | 7 |
| Stego App DB [8] | 24.120 | 26 |
| Wesaturate [7] | 3.648 | / |
| RAISE [5] | 8,156 | 3 |
| Dresden [6] | 1,491 | 73 (25 different models) |
| **Total** | **127,420** | **101** |

The LSSD database gathers RAW images available on the Internet, mostly from the Alaska#2 database [4] [3] to which we added images from the BOSS [1]

[3] Website of the ALASKA challenge#2: `https://alaska.utt.fr/`
Download page : `http://alaska.utt.fr/ALASKA_v2_RAWs_scripts.zip`.

[4], RAISE [6] [5], Dresden [10][6] and Wesaturate [7] databases, as well as StegoApp sites [14][8]. A total of 127,420 RAW images were collected. Table 1 lists the origin of the RAW images, while Figure 1 represents their distributions.

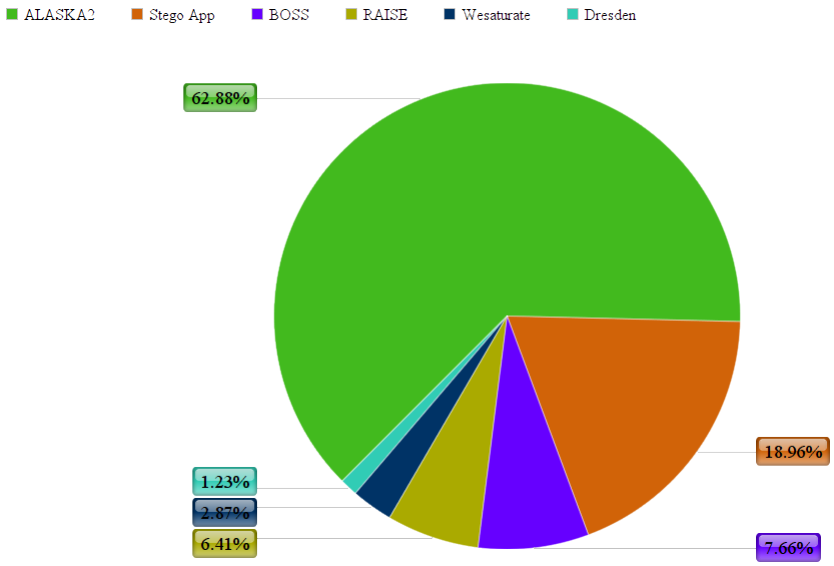


Fig. 1: Origin of RAW images in the LSSD database.

Note that the Alaska#2 database [4] covers a large variety of ISO parameters, ranging from 20 (used in general for smartphones) to 51,200 (only used for high-end devices). Among the 80,005 images of this database, 11,615 images have an ISO above 1,000 while 12,497 have an ISO below 100.

The majority of the databases reported in Table 1 are classically used by the community for steganalysis purpose. By combining them, we increase diversity and move to a more "real world" scenario [12]. We are thus closer to the "into the wild" spirit [4]. The ultimate goal is to reach the diversity findable when browsing the public images of the Internet and social networks.

[4] Challenge BOSS : `http://agents.fel.cvut.cz/boss/index.php?mode=VIEW&tmpl=about`
Download page : `ftp://mas22.felk.cvut.cz/RAWs`
[5] Obsolete download link `http://mmlab.science.unitn.it/RAISE/`.
[6] `http://forensics.inf.tu-dresden.de/ddimgdb`
Download page: `http://forensics.inf.tu-dresden.de/ddimgdb/selections`.
[7] Site closed on February 17, 2020 : `http://wesaturate.com/`.
[8] `https://data.csafe.iastate.edu/StegoDatabase/`.

### 2.2 The "development" pipeline

For the Alaska#2 competition, scripts were used to develop all the images according to some parameters [4]. It is thanks to these scripts that it was possible to obtain such a great diversity in the competition by playing on many parameters (see Table 2).

We apply these scripts to all the RAW images and we developed them into colour images (ppm format) whose size are $1024 \times 1024$ pixels (or slightly bigger). If the original image dimensions (width and height) are not equal, the colour image is cropped by taking its central part to get a $1024 \times 1024$ pixels image. A grey level image version is also generated using the standard luminance formula, transforming a RGB colour vector to a scalar representing the grey level:

$$grey\_value = 0.2989 \times R + 0.5870 \times G + 0.1140 \times B,$$

where $R, G, B \in [0, \ldots, 255]$ are the intensities of the red, green and blue channels. We will discuss in the next subsection the development parameters which were used.

As we want to get 2 million images, we add a process to multiply by 16 the number of images. Each colour (respectively the grey-level) ppm image is divided into 16 small images of size $256 \times 256$ pixels. Then, we run a compression of those 16 images, using the standard JPEG quantization matrices, with a quality factor of 75. The compression was carried out using the Python Imaging Library (PIL or *Pillow*)[9] package, version 1.1.7, which uses the plugin "JpegImagePlugin" to compress the images in the format $4:4:4$.

Figure 2 schematizes the steps of the complete development process.

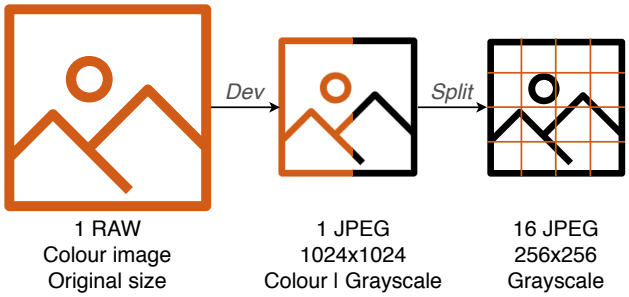


Fig. 2: Development process of a RAW image.

The $256 \times 256$ images have semantic content, a resolution and brightness variations that are close from those of images usually processed in steganalysis. Figure 3 (resp. Figure 4) shows two examples of JPEG grey-level (resp. colour) $256 \times 256$ images of the LSSD.

[9] Documentation: `https://pillow.readthedocs.io/en/stable/`.

(a) A developed image from the ALASKA database (number 3786).

(b) A developed image from the BOSS database (number 6456).

Fig. 3: Two $256 \times 256$ grey-scale images, after development process of the LSSD database.

(a) A developed image from the ALASKA database (number 51336).

(b) A developed image from the We-saturate database (index ZYlVRQY-DWE).

Fig. 4: Two $256 \times 256$ colour-scale images, after development process of the LSSD database.

It takes just under two days for all colour and grayscale images to be developed. The users of the LSSD database is free to either download the RAW images and redevelop those images or directly use the colour or grey-level JPEG $256 \times 256$ images.

### 2.3 Development parameters

In order to obtain the most realistic database, we have tested many development parameters (resize, crop, denoising, quality factor...). Table 2 summarizes all the parameters used during the generation of the dataset, reaching almost two million images.

All the processes explained below are done by using the *Rawtherapee*[10] v5.8 software which is a free, cross-platform raw image processing program.

Table 2: Parameters used in the image development process.

| Number | Name | Value |
|---|---|---|
| *1* | Demosaicking | Fast or DCB |
| *2* | Resize & Crop | Yes |
| | Taille (resize) | $1024 \times 1024$ |
| | Kernel (resize) | Nearest (0.2)<br>Bicubic (0.5)<br>Bilinear (0.3) |
| | Resize factor | depends on initial size |
| *3* | Unsharp Masking | No |
| *4* | Denoise<br>*(Pyramid Denoising)* | Yes |
| | Intensity | [0; 60] |
| | Detail | [0; 40] |
| *5* | Micro-contrast | Yes ($p = 0.5$) |
| *6* | Colour | No |
| *7* | Quality factor | 75 |

**1. Demosaicing:** Demosaicing is the process of reconstructing a full-resolution colour image from the sampled data acquired by a digital camera that applies a colour filter array to a single sensor (see, for example, the overview [13]). In the *Rawtherapee* software, we can find many demosaicing methods[11]. We selected the *DCB* method which produces similar results to the best method (AMaZE), plus the *Fast* method, based on nearest-neighbor interpolation, which is a lower-quality but very fast method. Notice that another available method called IGV is known to produce the most challenging images to steganalyze [4]. For each image, we select either the Fast or the DCB demosaicing algorithm with a probability of 35% and 65%, respectively. Demosaicked images are then saved in 16-bit TIFF format using the Python library *Pillow*[9].

**2. Resize & Crop:** the image is proportionally resized to final dimensions which are closest to $1024 \times 1024$ pixels as we will divide the resulting image into 16

[10] Software available at: `http://rawtherapee.com`
More information can be found at: `http://rawpedia.rawtherapee.com`

[11] Documentation about the different mosaicking methods of *Rawtherapee* can be found at: `https://rawpedia.rawtherapee.com/Demosaicing`

small images thereafter. If the image is not square, then we crop its center part, assuming that we will keep its semantic content. Resizing is performed using different kernels mentioned in Table 1, and it would have also been possible to use the $8 \times 8$ Lanczos filters.

**3. Unsharp Masking (USM):** After resizing, the image can eventually be sharpened. The USM process allows increasing the apparent acutance (edge contrast) of an image, making it appear clearer, even though it technically does not really sharpen the image. This process can be disabled. Note that for the *learning* database, the USM has been switched off. USM can be switched on for the development of the *test* database; this to introduce strong cover-source mismatch. More information about USM can be found in Appendix A.1 of [4].

**4. Denoising:** When the USM is switched off, denoising is systematically performed using a Pyramid Denoising based on wavelet decomposition. The denoising *intensity* parameter follows a gamma distribution with the pdf $P(x, a) = 10 \times \frac{x^{a-1} \exp(-x)}{\Gamma(a)}$, with $a = 4$, and rectified to belongs to $[0, 100]$. It controls the power of the noise reduction. The *Detail* parameter follows a uniform distribution $\mathcal{U}(\{0..60\})$, and it controls the restoration of textures in the image due to excessive noise. More information about Pyramid Denoising can be found in [4].

**5. Micro-contrast:** Since USM can generate artefacts, it is possible to apply a micro-contrast process. The micro-contrast process is performed after denoising with a probability of $p = 0.5$. This process is controlled by two parameters. The *strength* parameter follows a gamma distribution with the pdf $P(x, a) = 100 \times 0.5 \times \frac{x^{a-1} \exp(-x)}{\Gamma(a)}$, with $a = 1$, rectified on $[0, 100]$. This parameters allows to change the strength of the sharpness. The other parameter is the *uniformity* for the microcontrast enhancement. The uniformity follows the law $\lfloor \mathcal{N}(30, 5) \rfloor$ rectified on $[0, +\infty[$. That information is recalled in Appendix A.3 [4].

### 2.4 Choice of the JPEG quality factor

The Quality Factor (QF) of JPEG images is an essential element in the development pipeline. This factor can vary between 0 and 100, with 0 being a very poor quality, 50 being the minimum for good quality and 100 being the best possible. These quality factors are associated with $8 \times 8$ quantization matrices that are used in DCT image compression. There are typical (standard) matrices used in JPEG, but it is also possible to design ad-hoc quantization matrices (non-standard). In our case, we only use standard matrices; it results in a lower diversity compared to databases such as ImageNet [7]. Nevertheless, in future work, we would like to integrate this diversity to get as close as possible to real-world images and use image databases like ImageNet.

### 2.5 Reflection about quantization matrix diversity

For LSSD, we chose the quality factor $Q = 75$ (see table 2) with a standard quantization matrix. If we would use a JPEG database such as ImageNet [7], and desired to generate a database with a $Q$ "around" 75, we would have to

recompress all the images with a factor "equivalent" to $Q = 75$. In that case, it is possible to assume that a majority of JPEG images have a factor greater than 75. By recompressing at a lower factor, we would not introduce recompression artefacts. This new uncontrolled "real world" base would exhibit statistics of natural JPEG images (i.e. not recompressed), which would resemble those of LSSD, and the performances obtained could be compared with our "controlled" LSSD base.

Note that it is possible to recover, in the EXIF metadata, the quantization matrices (for each, Y, Cr, Cb, channel) of each JPEG image. With this information, it is easy to identify whether the matrices are standard or non-standard. As recalled in the article of Yousfi and Fridrich [19], the standard formula for obtaining a quantization matrix whatever the channel, given the Quality Factor, $Q$, is:

$$\mathbf{q}(Q) = \begin{cases} \max\left\{\mathbf{1}, \text{round}\left(2\left(1 - \frac{Q}{100}\right) \cdot \mathbf{q}(50)\right)\right\} & if\ Q > 50 \\ \min\left\{\{255 \cdot \mathbf{1}, \text{round}\left(\frac{50}{Q} \cdot \mathbf{q}(50)\right)\right\} & if\ Q \leq 50, \end{cases} \tag{1}$$

$\mathbf{q}(50)$ being the standard quantization matrix for $Q = 50$.

Re-compressing a JPEG image (coming from ImageNet) to a Quality Factor "close" to $Q = 75$, can be done by first computing the $\mathbf{q}(75)$, and then re-compress the input JPEG image using the $\mathbf{q}(75)$.

In the case of a non-standard JPEG input image, if we apply this process, we are losing the quantization diversity. An approach that would preserve this quantization diversity would be to find non-standard matrices noted $\mathbf{q}^{(ns)}(75)$, for the re-compression.

To do that, on can first estimate the non standard Quality Factor $Q^{(ns)}$ of the input JPEG image (trough iterative tests using the distance defined in Equation 8 of [19]), then compute the $\mathbf{q}^{(ns)}(50)$ by multiplying the quantization matrices by the pre-computed "passage"matrices, from $\mathbf{q}(Q^{(ns)})$ to $\mathbf{q}(50)$, and finally re-use equation 1 with substituting $\mathbf{q}(50)$ by $\mathbf{q}^{(ns)}(50)$, this in order to obtain $\mathbf{q}^{(ns)}(75)$.

The creation of an ImageNet database re-compressed to Q=75 is postponed to future work.

In conclusion, the diversity of the LSSD can be increased by increasing the development parameters range, by using additional development algorithms, by using various quality factors, and by using non-standard JPEG quantization matrices. Besides, in practice, the diversity of a JPEG colour image can also be increased compared to grey-scale images by using the following various formats: $4:4:4$, $4:2:2$, $4:2:0$ or $4:1:1$.

## 3 Application to DL-based steganalysis

In image classification, a field in which steganalysis is included, it is necessary to learn the neural network used on a training database and then observe its performance on a test database. The images in these bases must absolutely be

distinct. The interest of distinguishing these bases is to verify that the network is capable of learning and generalizing the information from the training base to get the best performance from images that it has never analyzed.

The article of Giboulot *et al.* [9], studying the effects of Unsharp Masking (see in section 2.3), pointed out that USM creates a strong mismatch phenomenon when used in the test set. For this reason, we decided to remove this processing when creating the learning database. The users can thus create a test database, using the USM process, and thus allowing the creation of cover-source mismatch phenomenon, that could be used in order to evaluate the impact of cover-source mismatch on steganalysis. Note that we also suppressed a few other processes such as some demosaicing algorithms and some resizing kernels when creating the learning database.

### 3.1 Training database construction

The RAW database consists of 127,420 images (see Table 1). We want to generate (from the RAW database) many learning datasets of different sizes from ten thousand to two million grey-scale JPEG images. One possible use is for evaluating the scalability of a steganalysis network, as in [15]. It is also necessary to set up a test dataset that will be the same for all learning datasets. This test dataset must be large enough to represent the diversity of developments, without being too disproportionate to the various sizes of the learning datasets. However, it should not be too large, to avoid high computational times in the test phase, even though during the test phase, calculations are faster.

We thus create several training datasets by, recursively, extracting a given number of images from the most extensive database (two million images). In total, we have six different sizes: 10k, 50k, 100k, 500k, 1M, 2M of cover images. So, when a database is used, it is important to take into account the corresponding stego images which doubles the total number of images. For example in the basis "LSSD_10k" there are 10,000 cover and 10,000 stego for a total of 20,000 images. In order to clearly identify the impact of increasing the size of the learning set, the smallest bases are included in the largest ones: $10k \subset 50k \subset 100k \subset 500k \subset 1M \subset 2M$. Each database tries to respect at best the initial ratio of the RAW images, which are shown in Table 3.

Table 3: Different LSSD database ratio with respect to the initial RAW image ratio.

| **Base name** | RAW | 100k-2M | 50k | 10k |
|---|---|---|---|---|
| ALASKA2 | 62.75% | = | = | +0.01% |
| BOSS | 7.84% | = | = | +0.01% |
| Dresden | 1.23% | = | = | +0.01% |
| RAISE | 6.40% | = | = | = |
| Stego App DB | 18.92% | = | = | = |
| Wesaturate | 2.86% | = | -0.01% | -0.03% |

### 3.2 Test database creation

We chose to generate a test set of one hundred thousand images. To this end, we isolated 6,250 RAW images with a distribution almost identical to the RAW image database (see section 2.1). These images will then undergo the same development as the one shown in Table 2. Note that this RAW test dataset, which is isolated from the training database, allows generating several different test datasets uncorrelated with the JPEG grey-scale image training dataset. Indeed, it is possible to use other development types, with different parameters, to introduce more or less mismatch. In particular, it is possible to incorporate the USM, which produces a strong mismatch and has a significant impact on network performance during the test phase [9].

Table 4: Images distribution of the original database in the test set.

| **Database name** | **Number of images** | **Percentage** | **RAW** |
|---|---|---|---|
| ALASKA2 | 3 970 | 63.52% | 62.75% |
| Stego App DB | 1 197 | 19.15% | 18.92% |
| BOSS | 496 | 7.94% | 7.84% |
| RAISE | 404 | 6.46% | 6.40% |
| Wesaturate | 183 | 2.93% | 2.86% |
| Dresden | 0 | 0% | 1.23% |

Table 4 lists the number of images and the percentage of each database used to form the shared test dataset. Images from Dresden have not been included in order to create a weak "mismatch" between learning and testing datasets. This phenomenon can be likened to a "real world" behaviour when the network learns on images that may not be seen again during the test phase.

### 3.3 Format of images

This database was used to make a test on scalability of a network in [15]. In this work, we applied the algorithm J-UNIWARD developed by Holub *et al.* [11] with a payload of 0.2 bpnzacs (bits per non zero AC coefficients). When Deep Learning is used, it is not possible to give images to the network in JPEG format, so they must be decompressed in MAT format.

Decompressed images are nevertheless much larger than the JPEG images. For example, for a $256 \times 256$ grey-scale image, its size is slightly more than 500 kB because it is stored in double format (the decompressed version is not rounded). Then, a database with almost four million images (cover and stego) takes more than 2 TB. When all data are combined (RAW images, JPEG colour cover, JPEG grey cover, JPEG Grey stego, MAT grey cover and MAT grey stego), we get almost 13 TB of data!

## 4 Conclusion

The main goal of this work is to provide to the community many controlled databases and a methodology adapted to steganalysis that allows learning on a large scale to get closer to real-world images diversity. The LSSD basis is available on the following website: `http://www.lirmm.fr/~chaumont/LSSD.html`

It is already possible to identify the first technical challenges when it comes to processing millions of images, such as the embedding time, the storage space required for a decompressed base. Furthermore, it is required to have scripts significantly optimized to create a new database; otherwise, these times quickly become excessive.

This new public repository gives the community many tools in order to better control their learning. The databases made of few thousand to multiple millions of images, already developed or re-developable is unique in the field. Moreover, the LSSD website is freely accessible, and additionally stores famous RAW databases for conservation since almost half of the RAW images present on the website are no longer downloadable on the Internet. By putting this new database online, it offers the community the possibility to diversify and broaden their research as they wish.

Note that we also generated the colour JPEG images, and those are also downloadable on our website. The studying of colour steganography and colour steganalysis is indeed a hot topic which has recently been addressed during Alaska#1 [4] [17] and Alaska#2 [5], [18] [3].

## Acknowledgment

The authors would like to thank the French Defense Procurement Agency (DGA) for its support through the ANR Alaska project (ANR-18-ASTR-0009). We also thank IBM Montpellier and the Institute for Development and Resources in Intensive Scientific Computing (IDRISS/CNRS) for providing us access to High-Performance Computing resources.

## References

1. Bas, P., Filler, T., Pevný, T.: 'Break Our Steganographic System': The Ins and Outs of Organizing BOSS. In: Proceedings of 13th International Conference on Information Hiding, IH'2011. Lecture Notes in Computer Science, Springer, vol. 6958, pp. 59–70. Prague, Czech Republic (May 2011)
2. Chaumont, M.: Deep Learning in steganography and steganalysis. In: Hassaballah, M. (ed.) Digital Media Steganography: Principles, Algorithms, Advances, chap. 14, pp. 321–349. Elsevier (Jul 2020)
3. Chubachi, K.: An Ensemble Model using CNNs on Different Domains for ALASKA2 Image Steganalysis. In: Proceedings of the IEEE International Workshop on Information Forensics and Security, WIFS'2020. Virtual Conference due to Covid (Formerly New-York, NY, USA) (Dec 2020)

4. Cogranne, R., Giboulot, Q., Bas, P.: The ALASKA Steganalysis Challenge: A First Step Towards Steganalysis. In: Proceedings of the ACM Workshop on Information Hiding and Multimedia Security. pp. 125–137. IH&MMSec'2019, Paris, France (Jul 2019)
5. Cogranne, R., Giboulot, Q., Bas, P.: Challenge Academic Research on Steganalysis with Realistic Images. In: Proceedings of the IEEE International Workshop on Information Forensics and Security, WIFS'2020. Virtual Conference due to Covid (Formerly New-York, NY, USA) (Dec 2020)
6. Dang-Nguyen, D.T., Pasquini, C., Conotter, V., Boato, G.: RAISE – A Raw Images Dataset for Digital Image Forensics. In: Proceedings of ACM Multimedia Systems. Portland, Oregon (March 2015)
7. Deng, J., Dong, W., Socher, R., Li, L.J., Li, K., Fei-Fei, L.: ImageNet: A Large-Scale Hierarchical Image Database. In: Proceedings of IEEE Conference on Computer Vision and Pattern Recognition, CVPR'2009. pp. 248–255 (2009)
8. Fridrich, J.: Steganography in Digital Media. Cambridge University Press (2009)
9. Giboulot, Q., Cogranne, R., Borghys, D., Bas, P.: Effects and Solutions of Cover-Source Mismatch in Image steganalysis. Signal Processing: Image Communication p. 115888 (Aug 2020)
10. Gloe, T., Böhme, R.: The 'Dresden Image Database' for Benchmarking Digital Image Forensics. In: Proceedings of the 25th Symposium On Applied Computing (ACM SAC 2010). vol. 2, pp. 1585–1591 (2010)
11. Holub, V., Fridrich, J., Denemark, T.: Universal Distortion Function for Steganography in an Arbitrary Domain. EURASIP Journal on Information Security, JIS (2014)
12. Ker, A.D., Bas, P., Böhme, R., Cogranne, R., Craver, S., Filler, T., Fridrich, J., Pevný, T.: Moving Steganography and Steganalysis from the Laboratory into the Real World. In: Proceedings of the 1st ACM Workshop on Information Hiding and Multimedia Security, IH&MMSec'2013. pp. 45–58. Montpellier, France (June 2013)
13. Menon, D., Calvagno, G.: Color image demosaicking: An overview. Signal Processing: Image Communication (8), 518 – 533 (2011)
14. Newman, J., Lin, L., Chen, W., Reinders, S., Wang, Y., Wu, M., Guan, Y.: StegoAppDB: a Steganography Apps Forensics Image Database. In: Proceedings of Media Watermarking, Security, and Forensics, MWSF'2019, Part of IS&T International Symposium on Electronic Imaging, EI'2019. Ingenta, Burlingame, California, USA (Jan 2019)
15. Ruiz, H., Chaumont, M., Yedroudj, M., Oulad-Amara, A., Comby, F., Subsol, G.: Analysis of the Scalability of a Deep-Learning Network for Steganography "Into the Wild". In: Proceeding of the 25th International Conference on Pattern Recognition, ICPR'2021, Worshop on MultiMedia FORensics in the WILD, MMForWILD'2021, Lecture Notes in Computer Science, LNCS, Springer. Virtual Conference due to Covid (Formerly Milan, Italy) (Jan 2021), `http://www.lirmm.fr/~chaumont/LSSD.html`
16. Yedroudj, M., Chaumont, M., Comby, F.: How to Augment a Small Learning Set for Improving the Performances of a CNN-Based Steganalyzer? In: Proceedings of Media Watermarking, Security, and Forensics, MWSF'2018, Part of IS&T International Symposium on Electronic Imaging, EI'2018. p. 7. Burlingame, California, USA (28 January - 2 February 2018)
17. Yousfi, Y., Butora, J., Fridrich, J., Giboulot, Q.: Breaking ALASKA: Color Separation for Steganalysis in JPEG Domain. In: Proceedings of the ACM Workshop

on Information Hiding and Multimedia Security. pp. 138–149. IH&MMSec'2019, Paris, France (Jul 2019)
18. Yousfi, Y., Butora, J., Khvedchenya, E., Fridrich, J.: ImageNet Pre-trained CNNs for JPEG Steganalysis. In: Proceedings of the IEEE International Workshop on Information Forensics and Security, WIFS'2020. Virtual Conference due to Covid (Formerly New-York, NY, USA) (Dec 2020)
19. Yousfi, Y., Fridrich, J.: JPEG Steganalysis Detectors Scalable With Respect to Compression Quality. In: Proceedings of Media Watermarking, Security, and Forensics, MWSF'2020, Part of IS&T International Symposium on Electronic Imaging, EI'2020. p. 10. Burlingame, California, USA (Jan 2020)
20. Zeng, J., Tan, S., Li, B., Huang, J.: Large-Scale JPEG Image Steganalysis Using Hybrid Deep-Learning Framework. IEEE Transactions on Information Forensics and Security (5), 1200–1214 (May 2018)