Analysis of the Scalability of a Deep-Learning Network for Steganography "Into the Wild"

Hugo Ruiz, Marc Chaumont, Mehdi Yedroudj, Ahmed Oulad-Amara, Frédéric Comby, Gérard Subsol

ICPR'2021, International Conference on Pattern Recognition, MMForWILD'2021, Worshop 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). Minor corrections added to the published version (29th of December 2020).

Analysis of the Scalability of a Deep-Learning Network for Steganography "Into the Wild"

Hugo Ruiz[1 [0000−0002−2806−2428]], Marc Chaumont[1,2 [0000−0002−4095−4410]], Mehdi Yedroudj[1 [0000−0001−6404−3876]], Ahmed Oulad Amara[1], 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, marc.chaumont, mehdi.yedroudj ahmed.oulad-amara, frederic.comby, gerard.subsol}@lirmm.fr
[2] University of Nîmes, France

Abstract. Since the emergence of deep learning and its adoption in steganalysis fields, most of the reference articles kept using small to medium size CNN, and learn them on relatively small databases.
Therefore, benchmarks and comparisons between different deep learning-based steganalysis algorithms, more precisely CNNs, are thus made on small to medium databases. This is performed without knowing:

1. if the ranking, with a criterion such as accuracy, is always the same when the database is larger,
2. if the efficiency of CNNs will collapse or not if the training database is a multiple of magnitude larger,
3. the minimum size required for a database or a CNN, in order to obtain a better result than a random guesser.

In this paper, after a solid discussion related to the observed behaviour of CNNs as a function of their sizes and the database size, we confirm that the error's power-law also stands in steganalysis, and this in a border case, i.e. with a medium-size network, on a big, constrained and very diverse database.

Keywords: Steganalysis · scalability · million images · "controlled" development.

1 Introduction

Steganography is the art of concealing information in a common medium so that the very existence of the secret message is hidden from any uninformed observer. Conversely, steganalysis is the art of detecting the presence of hidden data in such mediums [11].

Since 2015, thanks to the use of Deep-Learning, steganalysis performances have significantly improved [6]. Nevertheless, in many cases, those performances depend on the size of the learning set. It is indeed commonly shared that, to a certain extent, the larger the dataset, the better the results [24]. Thus, increasing the size of the learning set generally improves performance while also allowing for more diverse examples during training.

The objective of this article is to highlight the performance improvement of a Deep-Learning based steganalysis algorithm as the size of the learning set increases. In such a context, the behaviour of the network has never been studied before, and numerous questions related to model size and dataset size are still unsolved.

In section 2, we discuss those questions and the generic laws or models that have been proposed by the scientific community. Next, in Section 3, we present the testbench used to assess the error power-law. We justify and discuss the various choices and parameters setting, required in order to run the experiments. We also present the Low Complexity (LC-Net) network [15] which is a CNN that was considered as the state of the art algorithm for JPEG steganalysis in 2019 and 2020. In the experimental section 4, we briefly present the Large Scale Steganalysis Database (LSSD) [18], the experimental protocol and describe the conducted experiments. We then analyze the accuracy evolution with respect to the learning set size, and predict, thanks to the error power-law, the reachable efficiency for very big databases. Finally, we conclude and give some perspectives.

2 Model scaling and Data scaling

Many theoretical and practical papers are trying to better understand the behaviour of neural networks when their dimension is increasing (the depth or the width) [4] [20] [2] [16], or when the number of examples is increasing [13] [19] [17]. To this end, lots of experiments are done in order to observe the evolution of the test error as a function of the *model size*, or as a function of the *learning set size* and general laws are proposed for modelling the phenomenon. Those are essential researches because finding some generic laws could confirm that CNN users are applying the right methodologies.

Even if they have access to a large dataset, which is, in many domains, rarely possible, CNN users may have to restrict the learning to small to medium database, and small to medium-size models, during the preliminary experiments. Then, once satisfied, if possible, they can run a long time learning on a large dataset (i.e. greater than 10^6 examples) and eventually with a large network (i.e. greater than 10^6 parameters).

The questions arising by users are then: do the comparisons between various models, when evaluated on a small dataset, also stand when the dataset size increase. In other words, can we reasonably conclude on the best model when comparing the networks on a small dataset? What is the behaviour of a medium-size network when the dataset size increases? More generally, is there a collapse in performance when a model or the dataset scales up? Or, will the accuracy increase? Should we prefer bigger models? Is there a minimum required size for models or dataset?

In the studies related to the *model scaling*, researchers have observed three regions depending on the model size. There is the *underfitting*, the *overfitting*, and finally the *over-parameterized* region. The transition point to the over-

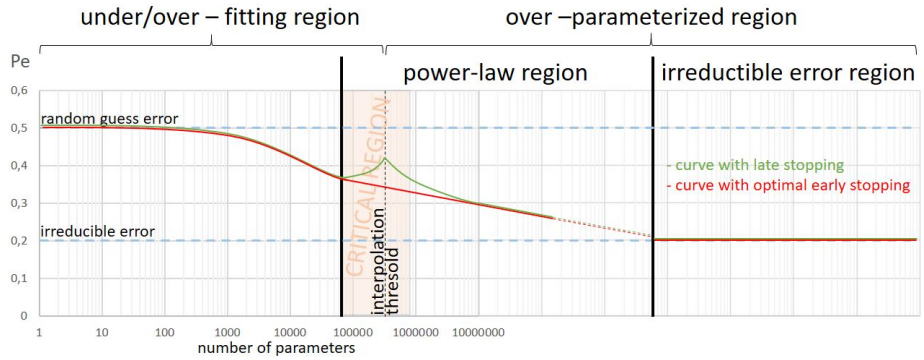

Fig. 1: Schematic generic evolution of the test error depending on the model size.

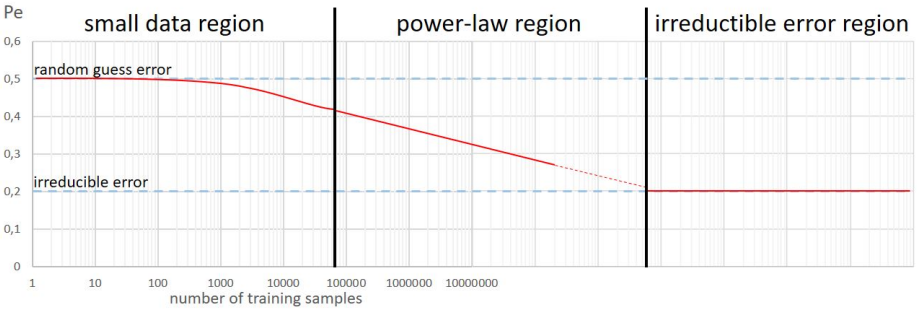

Fig. 2: Schematic generic evolution of the test error depending on the dataset size.

parameterized region is named the *interpolation threshold*. Figure 1 shows those three regions (note that the abscissa scale is logarithmic).

When looking to the curve of the error as a function of the model size (the green curve; this without optimal early stopping), we can observe a *double descent* [16]. In general, a practical conclusion is that the over-parameterized networks, i.e. with millions of parameters, can be used for any task and that it is beneficial using those more complex models. This idea has been, for example, used in practice in order to build gradually larger EfficientNet networks [21]. Note that this network has been strongly used by competitors [27] [7] during the Alaska#2 competition [10].

In the studies related to the data scaling, researchers have observed that there are also three regions depending on the dataset size [13]. There is the *small data* region, the *power-law* region, and finally the *irreducible error* region. Figure 2 shows those three regions (note that the abscissa scale is logarithmic).

In the power-law region, the more data, the better results [19],[23].

Recently, the authors of [17] have proposed a generic law that models the behaviour when scaling both the model size and the dataset size. Briefly, the

test error noted ϵ is expressed as the sum of two exponentially decreasing term plus a constant. The first term is function of the dataset size, noted n, and the second one is function of the model size, noted m [17]:

$$\begin{aligned} \epsilon : \mathbb{R} \times \mathbb{R} &\rightarrow [0,1] \\ \epsilon(m,n) &\rightarrow \underbrace{a(m)n^{-\alpha(m)}}_{dataset\ power-law} + \underbrace{b(n)m^{-\beta(n)}}_{model\ power-law} + c_\infty \end{aligned} \tag{1}$$

with $\alpha(m)$ and $\beta(n)$ controlling the rate of the error decrease, depending on m and n respectively, and c_∞ the irreducible error, a real positive constant, independent of m and n.

Then, the authors propose a simplification of the expression in:

$$\tilde{\epsilon}(m,n) = an^{-\alpha} + bm^{-\beta} + c_\infty \tag{2}$$

with a, b, α, and β real positive constants, and then use a complex envelope function in order to represent the transition from the *random-guess error* region to the *power-law* region [17]: $\hat{\epsilon}(m,n) = \epsilon_0 \left\| \tilde{\epsilon}(m,n)/(\tilde{\epsilon}(m,n) - i\eta) \right\|$, with $\epsilon_0 = 1/2$ for balanced binary classification, $i = \sqrt{-1}$, and $\eta \in \mathbb{R}$.

The interesting aspect with this function is that once the parameters a, b, α, β, and η are learnt using a regression on experimental points, obtained at various m and n values, with m and n not too high, one can answer to the questions mentioned above [3].

Now, let us go back to a more practical aspect. Suppose we are learning with an efficient network with enough parameters, i.e. on the right region relative to the interpolation threshold, possibly leaving the critical region (see Figure 1), and use a data-set of medium size such that we are no more in the small data region, avoiding us a random guess error (see Figure 2). When studying the effect of increasing the data on the error, we should be in the power-law region and equation 2 can be simplified, as in [13]:

$$\epsilon(n) = a'n^{-\alpha'} + c'_\infty \tag{3}$$

In the rest of our paper, we are observing, in the context of JPEG steganalysis, the behaviour of a medium network when the dataset size increases. Then, we confront these results to the power-law related to the data scaling (Equation 3). Moreover, we are checking a "border case" because we are using a medium size model (3.10^5 parameters), and because we are using a very diverse database (the LSSD database [18] derived, from a part, from Alaska#2 [10]). This could result in a collapse in performance as the database increases, and failure to comply with the evolution law of the estimation error.

[3] See the paper [17], and the discussions here: `https://openreview.net/forum?id=ryenvpEKDr`.

3 A test bench to assess scalability for DL-based steganalysis

3.1 Discussion on the test bench design

Choice of the network: Our objective is to evaluate the accuracy (or equivalently the probability of error) as a function of the increase in the size of the dataset. But, as many researchers, we are limited by computational resources, so we need a low complexity network. We thus selected the Low Complexity network (LC-Net) [15], for its medium size (300 000 parameters), and its good performance as it is recognized as a state-of-the-art CNN for JPEG steganalysis at the date we ran the experiments (between September 2019 to August 2020). Note that we can consider that the LC-Net is probably close to the *interpolation threshold*[4] which implies that we must take caution to do an early stopping, close to the optimal, during the learning phase.

Choice of the payload: Another critical thing is that the network should be sufficiently far from the random-guess region in order to observe a concrete improvement of the performance when scaling the database. So we have to choose a payload in order that the LC-Net accuracy is quite far from 50%. This is quite challenging because there are no experiments results for LC-Net [15] on "controlled" databases such as LSSD having a large diversity. More generally, there are not so many experiments reported before the summer of 2020 that used the unique controlled and diverse, Alaska#1 [8] database. The objective, here, was to obtain accuracy between 60% and 70% for a small database (but not too small [5]), in order to observe progression when the dataset is scaled and to let the possibility to future better networks to beat our results with sufficient margin. After a lot of experimental adjustments, either related to the building of the LSSD database [18] or related to the LC-Net [15], we found that 0.2 bpnzacs was a good payload for grey-level JPEG 256×256 image with a quality factor of 75, ensuring to be quite far from the random-guess region for a small database of 20,000 images made of cover and stego images.

Choice related to the database: We decided to work on grey-level JPEG images in order to put aside the color steganalysis, which is still recent and not enough theoretically understood [1]. Related to color steganalysis, the reader can consult the paper WISERNet [29] or the description of the winning proposition for Alaska#1 [26]. The reader can also read the even more recent papers, in

[4] The ResNet18 with a width=10 is made of 300,000 parameters and is in the *interpolation threshold* region for experiments run on CIFAR-10 and CIFAR-100 in [16].

[5] A too small database could bias the analysis since there is a region where the error increases when the dataset increase (see [16]). For example, in [23], we report that the number of images needed, for the medium size Yedroudj-Net [25], to reach a region of good performance for spatial steganalysis (that is the performance of a Rich Model with an Ensemble Classifier), is about 10,000 images (5,000 covers and 5,000 stegos) for the learning phase, in the case where there is no cover-source mismatch, and the image size is 256×256 pixels.

the top-3 of Alaska#2 [27], [7], which are based on an ensemble of networks (for example EfficientNet [21]), which have preliminary learned on ImageNet. Related to steganography, the most recent proposition in order to take into account the three channels during the embedding can be found in [9] and was used in order to embed payload in Alaska#2 images.

We also decided to work only on the quality factor 75, and thus let apart the quantization diversity. Nevertheless, the conclusions obtained in the following could probably extend to a small interval around quality factor 75. Indeed, the authors of [28] show that applying a steganalysis with a JPEG images database made of multiple quality factors could be done without efficiency reduction, using a small set of dedicated networks, each targeting a small interval of quality factor. Finally, it is maybe possible to use a unique network when there are various quality factors, as it has been done by a majority of competitors during Alaska#2, thanks to the use of a pre-learned network on ImageNet. An extension of our work to a database with a variety of quality factors is postponed to future research.

Another reason to work with a quality factor of 75 is that we had in mind, for future work, the use of non controlled images such as ImageNet. ImageNet is made of JPEG compressed images whose development process is not controlled and whose quality factors are multiple. By re-compressing the images with a smaller quality factor, such as 75, the statistical traces of the first initial compression are removed. Such a re-compression would allow us to work on images with a roughly similar quality factor, and whose statistical properties would not be too poor. Additionally, the experimental methodology would be close to those exposed in the current paper and would facilitate comparisons.

Finally, we built the LSSD database [18] in order to have a proper set for our experiments. For this database, we used controlled development using scripts inspired by Alaska#2. The LSSD was obtained by merging 6 public RAW images databases including Alaska#2. Without being as varied as images available on the Internet, we consider that diversity is nevertheless significant. There are 2 million covers in LSSD learning database, and we built "included subsets", from those 2 million covers in order to run our experiments.

With all those precautions, at the date where experiments started i.e. before running the experiment with an increasing number of examples, we assumed that we were at the border case, where the power-law on the data is valid.

3.2 Presentation of LC-Net

In this paper, we use the Low Complexity network (LC-Net) [15], which is a convolutional neural network proposed in 2019 for steganalysis in the JPEG domain. Its design is inspired by ResNet [12], the network that won the ImageNet competition in 2015. LC-Net performance is close to the state-of-the-art SRNet [5], with the advantage of a significant lower complexity in terms of the number of parameters [15] (twenty times fewer parameters than SRNet). This reduction in the number of parameters leads to less learning time as it converges faster toward an optimal solution.

LC-Net is composed of three modules: pre-processing, convolution and classification (see Fig. 3).

The pre-processing module has a total of 4 convolutional layers, with the first layer kernels initialized using 30 SRM filters. These high-pass filters are commonly used in steganalysis [22], [25]. They allow the network to reduce its learning time but also to converge when using a small learning set. For instance, using the BOSS database [3], only 4,000 pairs of cover-stego images may be sufficient to perform learning "from scratch" and get good performance [23]. The parameters of the first layer are not fixed; they are optimized during training. This first layer is followed by an activation function "TLU" (Truncated Linear Unit) [22], where the T threshold is set to 31. For the remaining layers of the network, the "ReLU" (Rectified Linear Units) activation function is used. Batch normalization is used. No pooling is applied in this first module in order to preserve the stego signal.

The convolutional module is composed of 6 blocks, all with residual connections. These connections allow to avoid the vanishing gradient phenomenon during the back-propagation and thus to have deeper networks. The first two blocks have only two convolutional layers with direct residual connections. Blocks 3 to 6 include 3×3 convolutions with a stride equal to 2 to reduce the size of the feature maps. Indeed, it preserves the complexity of the computation time per layer when the number of filters is doubled [12]. Blocks 4 to 6, are Bottleneck residual blocks [12]. These blocks include 3 convolutional layers, a 1×1 convolution layer, a 3×3 convolution layer and another 1×1 convolution layer. The use of the Bottleneck block [12] allows the Low Complexity network having fewer parameters.

Finally, the classification module consists of a "Fully Connected" (FC) layer and a Softmax activation function. This function allows for obtaining the probabilities of the cover and stego classes.

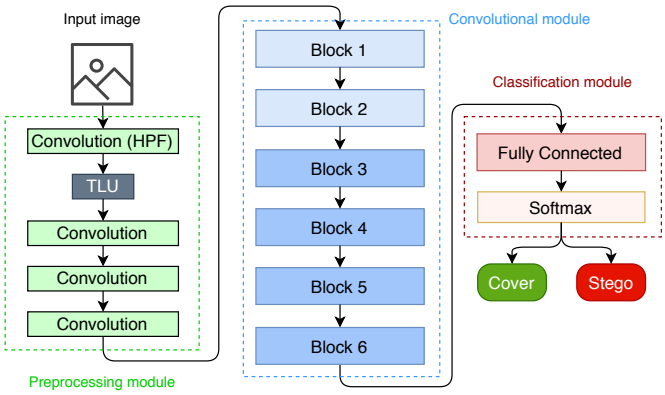

Fig. 3: Architecture of LC-Net

4 Experiments and results

4.1 Dataset and software platform

As mentioned previously, the experiments were conducted on the LSSD database [18][6]. We are using greyscale JPEG images with a quality factor of 75. The size of those JPEG images is 256×256 pixels. They were obtained by developing RAW images (data issued from the camera sensors) from ALASKA#2, BOSS, Dresden, RAISE, Stego App, and Wesaturate public databases. The development scripts are inspired from the Alaska#2 scripts.

The cover database used for the learning phase is made of 2 million images. There is also a version with 1M, 500k, 100k, 50k, and 10k images. The 1M cover images database is a subset of the 2 million one, and so on, until the 10k cover images database. Each of those cover databases retains the same proportions of images from the different public databases.

The cover database used for the test phase is made of 100k images and will always be the same whatever the experiments. This test database is obtained by developing RAW images which were not present in the learning cover database. The test cover database roughly keeps the distribution of the origins of the public databases. Thus, the steganalysis scenario is close to a clairvoyant scenario where the test set and learning set are statistically very close.

In our experiments, we have only used the 500k, 100k, 50k and 10k versions of the cover database due to excessively long learning time process for 1M and 2M images versions.

The study was conducted on an IBM software container, with access to 144 supported POWER9 Altivec processors (MCPs) and two GV100GL graphic cards (Tesla V100 SXM2 16Gb).

4.2 Training, validation, and testing

The embedding process has been done with the Matlab implementation of J-UNIWARD algorithm [14], with a payload of 0.2 bits per non-zero AC coefficient (bpnzacs). It took almost three days (2 days and 20 hours) for the embedding on an Intel Xeon W2145 (8 cores, 3.74.5 GHz Turbo, 11M cache).

Before feeding the neural network, JPEG images have to be decompressed in order to obtain spatial non rounded "real values" images. This essential step takes approximately 18 hours for all the images. Note that storage space requirement becomes important. Indeed, for a 256×256 grey-scale image, the file's size is around 500 kB when it is stored in MAT format in *double* format. Thus, a database of 2M images requires a storage space of about 2 TB, and the learning cover bases, from 10k to 2M images, as well as the test cover database, in both JPEG and MAT format, occupy 3.8 TB.

In order to avoid storing all the decompressed images, one would have to perform an "online" decompression asynchronously coupled with an "online"

[6] The LSSD database is available at: http://www.lirmm.fr/~chaumont/LSSD.html.

mini-batch build, in order to feed the neural network "on flight". Note that with such a solution it could be possible to accelerate the global learning time, by directly working with the GPU RAM, instead of the CPU RAM or the Hard Disk, which have longer access time. This "online" treatment is not an easy task to carry, and the problem will have to be addressed for databases exceeding tens or even hundreds of million images.

The training set is split into two sets with 90% for the "real" training set and 10% for validation. As said previously, the test set is always the same and is made of 200k images (cover and stegos).

4.3 Hyper-parameters

To train our CNN we used a mini-batch stochastic gradient descent without dropout. We used the majority of the hyper-parameters of the article [15]. The learning rate, for all parameters, was set to 0.002 and is decreased at the epoch 130 and 230, with a factor equal to 0.1. The optimizer is Adam, and the weight decay is 5e-4. The batch size is set to 100 which corresponds to 50 cover/stego pairs. In order to improve the CNN generalization, we shuffled the entire training database at the beginning of each epoch. First, layer was initialized with the 30 basic high-pass SRM filters, without normalization, and the threshold of the TLU layer equals 31 similarly to [22], [25]. We made an early stop after 250 epochs as in [15]. The code and all materials are available at the following link: http://www.lirmm.fr/~chaumont/LSSD.html

4.4 Results and discussion

The different learning sets, from 20k to 1M images (covers and stegos), were used to test the LC-Net. Table 1 gives the network performances when tested on the 200k test cover/stego images database. Note that several tests were conducted for each size of the learning set and the displayed accuracies represent an average computed on the 5 best models recorded during the training phase. Those 5 best models are selected thanks to the validation set.

Table 1: Average accuracy evaluated on the 200k cover/stego images test set, with respect to the size of the learning database.

Database size	**Nb. of tests**	**Accuracy**	**Std. dev.**	**duration**
20,000	5	62.33%	0.84%	2h 21
100,000	5	64.78%	0.54%	11h 45
200,000	5	65.99%	0.09%	23h 53
1,000,000	1	68.31%	/	10d to 22d

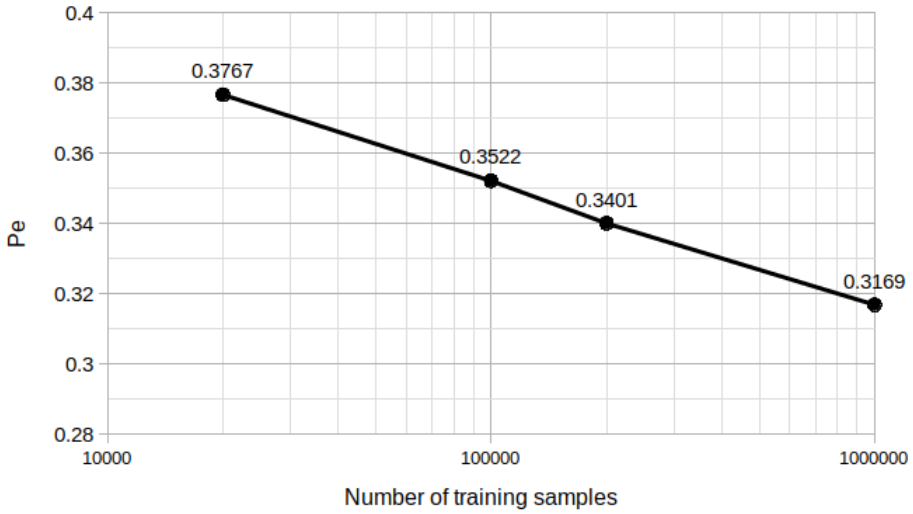

Fig. 4: Average probability of error with respect to the learning database size. Notice that the abscissa scale is logarithmic.

Before analyzing the network results, we should note that learning times become significant (from 10 days to 22 days [7]) once the number of images exceeds 1 million. This is a severe problem which did not allow us, due to lack of time, to run an evaluation on the 2 million (1M covers + 1M stegos) and 4 million (2M covers + 2M stegos) databases as the learning duration would have been higher than one month. Moreover, the hardware architecture would probably not be able to store all the database in RAM, and it is likely that the paging optimizations would be no longer valid on such a volume of data. The transfer of an image from hard disk to the RAM, and then to the GPU, becomes then the bottleneck in the learning process. As explained previously, to cope with this problem, a decompression thread and a reading thread with the use of a shared file and the use of semaphores could be used to reduce the memory storage and the transfer on the GPU memory. It would make it possible to build image mini-batches "on flight" during network learning.

Results of Table 1, obtained for the payload 0.2 bpnzacs, confirm that the larger the learning set is (100k, 200k, 1M), the better the accuracy is. For the 20k database, the accuracy is 62% and increases by almost 2% each time the size of the learning set is increased. Moreover, the standard deviation is getting smaller and smaller, which highlights that the learning process is more and more stable as the database increases.

[7] Experiments with 1M images were disrupted by a maintenance of the platform and it took 22 days. Nevertheless, we rerun the experiment on a similar platform, without any disruption in learning, and the duration was only 10 days. So we report both values to be more precise.

These first results would mean that most of the steganalysis experiments run by the community, using a medium size (but also a large size) Deep Learning network, are not done with enough examples to reach the optimal performance, since most of the time the database is around 10k (BOSS learning set) to 150k images (Alaska#2 learning set with only one embedding algorithm). As an example, in our experiments, the accuracy is already improved by 6% when the database increase from 20k to 1M images and the accuracy can probably be improved by increasing the dataset size since the irreducible error region is probably not reached.

Those results also confirm that a medium-size network such as LC-Net does not have its performance collapsing when the database size increases.

More interestingly, we observe (see Figure 4) an exponential decrease in the probability of error in the function of the dataset size. This is a direct observation of the power-law discussed in Section 2. Using a non-linear regression with Lagrange multipliers[8], we have estimated the parameters of Equation 3:

$$\epsilon(n) = 0.492415 n^{-0.086236} + 0.168059 \tag{4}$$

The sum of the square error is 4.4×10^{-6}. Since there are only four points for the regression, it is probably erroneous to affirm that the irreducible error is $c'_\infty = 16.8\%$. However, we can use equation 4 to predict without much error, that if we use 2M images for the learning, the probability of error will be close to 30.9%, and if we use 20M images, the probability of error will be again reduced of 2% and will be 28.3%. Note that if c_∞ was equal to 0, the probability of error would be 30.7% for 2M images, and 27.8% for 20M images. If we consider a probability of error of 28.3% for 20M of images, the gain obtained compared to the probability of error of 37.7% with 20k images, corresponds to 9% increase which is a considerable improvement in steganalysis domain.

To conclude, the error power-law also stands for steganalysis with Deep Learning, and this even when the networks are not very big (300,000 parameters), even when starting with a medium-size database (here, only 20k images), and even if the database is diverse (use of Alaska#2 development script and around 100 camera models). So, bigger databases are needed for optimal learning, and using more than one million images are likely needed before reaching the "irreducible error" region [13].

5 Conclusion

In this paper, we first have recalled the recent results obtained by the community working on AI, and related to the behaviour of Deep-Learning networks when the model size or the database size is increasing. We then proposed an experimental

[8] The initial point for the non-linear regression is set to $a' = 0.5$, $\alpha' = 0.001$ and $c'_\infty = 0.01$, with c'_∞ forced to be positive. The Matlab function is *fconmin* and the stop criterion is such that the mean of the sum of square error is under 10^{-6}.

setup in order to evaluate the behaviour of a medium-size CNN steganalyzer (LC-Net) when the database size is scaled.

The obtained results show that a medium-size network does not collapse when the database size is increased, even if the database is diverse. Moreover, its performances are increased with the database size scaling. Finally, we observed that the error power-law is also valid for steganalysis domain. We thus estimated what would be the accuracy of the network if the database would have been made of 20 million images.

Future work will require to be done on a more diverse database (quality factors, payload size, embedding algorithm, colour, less controlled database), and also other networks. More practically, an effort should be made in order to reduce the learning time, and especially memory management. Finally, there are still open questions to solve such as: finding a more precise irreducible error value, finding the slope of the power-law depending on the starting point of the CNN (use of transfer, use of curriculum, use of data-augmentation such as pixels-off [24]), or finding innovative techniques when the database is not huge in order to increase the performances.

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. Abdulrahman, H., Chaumont, M., Montesinos, P., Magnier, B.: Color Images Steganalysis Using RGB Channel Geometric Transformation Measures. Security and Communication Networks (15), 2945–2956 (2016)
2. Advani, M.S., Saxe, A.M., Sompolinsky, H.: High-Dimensional Dynamics of Generalization Error in Neural Networks. Neural Networks pp. 428–446 (2020)
3. 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)
4. Belkin, M., Hsu, D., Ma, S., Mandal, S.: Reconciling modern machine-learning practice and the classical bias–variance trade-off. Proceedings of the National Academy of Sciences (32), 15849–15854 (2019)
5. Boroumand, M., Chen, M., Fridrich, J.: Deep Residual Network for Steganalysis of Digital Images. IEEE Transactions on Information Forensics and Security (5), 1181 – 1193 (May 2019)
6. 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)

7. 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)
8. 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)
9. Cogranne, R., Giboulot, Q., Bas, P.: Steganography by Minimizing Statistical Detectability: The Cases of JPEG and Color Images. In: Proceedings of the ACM Workshop on Information Hiding and Multimedia Security. p. 161–167. IH&MMSec'2020 (Jun 2020)
10. 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)
11. Fridrich, J.: Steganography in Digital Media. Cambridge University Press (2009)
12. He, K., Zhang, X., Ren, S., Sun, J.: Deep Residual Learning for Image Recognition. In: Proceedings of IEEE Conference on Computer Vision and Pattern Recognition, CVPR'2016. pp. 770–778. Las Vegas, Nevada (Jun 2016)
13. Hestness, J., Narang, S., Ardalani, N., Diamos, G., Jun, H., Kianinejad, H., Patwary, M.M.A., Yang, Y., Zhou, Y.: Deep Learning Scaling is Predictable, Empirically. In: Unpublished - ArXiv. vol. abs/1712.00409 (2017)
14. Holub, V., Fridrich, J., Denemark, T.: Universal Distortion Function for Steganography in an Arbitrary Domain. EURASIP Journal on Information Security, JIS (2014)
15. Huang, J., Ni, J., Wan, L., Yan, J.: A Customized Convolutional Neural Network with Low Model Complexity for JPEG Steganalysis. In: Proceedings of the ACM Workshop on Information Hiding and Multimedia Security. pp. 198–203. IH&MMSec'2019, Paris, France (Jul 2019)
16. Nakkiran, P., Kaplun, G., Bansal, Y., Yang, T., Barak, B., Sutskever, I.: Deep double descent: Where bigger models and more data hurt. In: Proceedings of the Eighth International Conference on Learning Representations, ICLR'2020. Virtual Conference due to Covid (Formerly Addis Ababa, Ethiopia) (Apr 2020)
17. Rosenfeld, J.S., Rosenfeld, A., Belinkov, Y., Shavit, N.: A constructive prediction of the generalization error across scales. In: Proceedings of the Eighth International Conference on Learning Representations, ICLR'2020. Virtual Conference due to Covid (Formerly Addis Ababa, Ethiopia) (Apr 2020)
18. Ruiz, H., Yedroudj, M., Chaumont, M., Comby, F., Subsol, G.: LSSD: a Controlled Large JPEG Image Database for Deep-Learning-based Steganalysis "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`
19. Sala, V.: Power law scaling of test error versus number of training images for deep convolutional neural networks. In: Proceedings of the Multimodal Sensing: Technologies and Applications. vol. 11059, pp. 296 – 300. International Society for Optics and Photonics, SPIE, Munich (2019)

20. Spigler, S., Geiger, M., d'Ascoli, S., Sagun, L., Biroli, G., Wyart, M.: A jamming transition from under- to over-parametrization affects generalization in deep learning. Journal of Physics A: Mathematical and Theoretical (47), 474001 (oct 2019)
21. Tan, M., Le, Q.: EfficientNet: Rethinking Model Scaling for Convolutional Neural Networks. In: Proceedings of the 36th International Conference on Machine Learning, PMLR'2019. vol. 97, pp. 6105–6114. Long Beach, California, USA (Jun 2019)
22. Ye, J., Ni, J., Yi, Y.: Deep Learning Hierarchical Representations for Image Steganalysis. IEEE Transactions on Information Forensics and Security, TIFS (11), 2545–2557 (Nov 2017)
23. 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)
24. Yedroudj, M., Chaumont, M., Comby, F., Oulad Amara, A., Bas, P.: Pixels-off: Data-Augmentation Complementary Solution for Deep-Learning Steganalysis. In: Proceedings of the 2020 ACM Workshop on Information Hiding and Multimedia Security. p. 39–48. IHMSec '20, Virtual Conference due to Covid (Formerly Denver, CO, USA) (Jun 2020)
25. Yedroudj, M., Comby, F., Chaumont, M.: Yedrouj-Net: An Efficient CNN for Spatial Steganalysis. In: Proceedings of IEEE International Conference on Acoustics, Speech and Signal Processing, ICASSP'2018. pp. 2092–2096. Calgary, Alberta, Canada (Apr 2018)
26. 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. p. 138–149. IH&MMSec'2019, Paris, France (Jul 2019)
27. 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)
28. 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)
29. Zeng, J., Tan, S., Liu, G., Li, B., Huang, J.: WISERNet: Wider Separate-Then-Reunion Network for Steganalysis of Color Images. IEEE Transactions on Information Forensics and Security (10), 2735–2748 (2019)