Pick the Largest Margin for Robust Detection of Splicing

Julien Simon de Kergunic, Rony Abecidan, Patrick Bas, Vincent Itier

Pick the Largest Margin for Robust Detection of Splicing

Julien SIMON de KERGUNIC
Centrale Lille,
Cité Scientifique,
59650 Villeneuve-d'Ascq, France
Email: julien.simon@centrale.centralelille.fr

Rony Abecidan
Univ. Lille, CNRS, Centrale Lille,
UMR 9189 CRIStAL,
F-59000 Lille, France
Email: rony.abecidan@univ-lille.fr

Patrick Bas
Univ. Lille, CNRS, Centrale Lille,
UMR 9189 CRIStAL,
F-59000 Lille, France
Email: patrick.bas@cnrs.fr

Vincent Itier
IMT Nord Europe, Institut Mines-Télécom,
Centre for Digital Systems, Univ. Lille, CNRS, Centrale Lille,
UMR 9189 CRIStAL, F-59000 Lille, France
Email: vincent.itier@imt-nord-europe.fr

***Abstract*—Despite advancements in splicing detection, practitioners still struggle to fully leverage forensic tools from the literature due to a critical issue: deep learning-based detectors are extremely sensitive to their trained instances. Simple post-processing applied to evaluation images can easily decrease their performances, leading to a lack of confidence in splicing detectors for operational contexts. In this study, we show that a deep splicing detector behaves differently against unknown post-processes for different learned weights, even if it achieves similar performances on a test set from the same distribution as its training one. We connect this observation to the fact that different learnings create different latent spaces separating training samples differently. Our experiments reveal a strong correlation between the distributions of latent margins and the ability of the detector to generalize to post-processed images. We thus provide to the practitioner a way to build deep detectors that are more robust than others against post-processing operations, suggesting to train their architecture under different conditions and picking the one maximizing the latent space margin.**

I. Introduction

Splicing is the process of altering an image by taking objects from a different image and inserting them into the original, effectively changing its meaning or message. Modern splicing detectors leverage deep architectures harnessing noise anomalies to perform their detections [1], [2], [3], however their effectiveness in real-world scenarios often do not match the performances reported in the literature. One common explanation of this gap of performance is attributed to unknown post-processing transformations applied to the spliced images [4], [5]. A simple post-processing pipeline applied on all images under scrutiny is indeed enough to create a significant domain shift, altering the distribution of both pristine and manipulated images and disturbing detectors trained by forensic analysts. This generalization problem is due to the fact that post-processing operations (e.g. sharpening, denoising, resizing etc.) alter the noise distribution of images while creating dependencies between pixels of both original and forged areas.

WIFS'2024, December, 2-5, 2024, Roma, Italy. XXX-X-XXXX-XXXX-X/XX/$XX.00

Seed	Source accuracy	Mean target accuracy	Std. of Target Accuracy
4	84%	72%	1.8
6	84%	74%	2.0
8	84%	66%	2.3

TABLE I
IMPACT OF DIFFERENT INITIALIZATION SEEDS ON THE OUT-OF-DOMAIN PERFORMANCE OF THE BAYAR DETECTOR [1]. THE MEAN TARGET ACCURACY IS COMPUTED BY AVERAGING THE TEST ACCURACIES OF A BAYAR DETECTOR TRAINED WITH THREE DISTINCT SEEDS ON 20 POST-PROCESSED TARGETS PROCESSED WITH RAWTHERAPEE. (TRAIN) $N_{source} \sim 20,000$ PATCHES; (TEST) $N_{source} \sim N_{target} \sim 7,000$ PATCHES.

Postprocessing consequently makes forgeries less noticeable for forensic detectors designed to detect statistical anomalies between pixels. Hence, all forgery detectors may be affected by post-processing.

A. Robust Detection against Post-Processing: Prior Arts

In forensic literature, the *domain shift* caused by post-processing pipelines is present in various manipulation detection problems, including photo-editing & watermarking detection, and steganalysis [4], [6], [7]. Several solutions, often inspired by the machine learning literature, have been proposed to address this issue.

- *Data-centric* approaches focus on searching or building relevant training sets allowing any detector to generalize across multiple distributions which are different from the training domain. The simplest method involves artificially augmenting the training set by randomly applying standard post-processing operations (denoising, sharpening, JPEG compression, etc.) to the source images [8]. However, recent studies in steganalysis suggest that it is more effective to carefully select the operations to apply rather than using a random mixture of operations [9], [10]. When target images are available, other strategies propose to select relevant sources for the target or to estimate the post-processing pipeline applied to the target to reproduce training samples following the target distribution [11], [12]. This allows to train a detector on a source very related to the target. A common issue with data-centric approaches is the uncertainty that the constructed or selected datasets adequately cover all target domains of interest.

- *Detector-centric* approaches aim to build or update detectors to be more robust against out-of-distribution data. This is particularly relevant in domain adaptation, where a model trained on a labeled source needs to generalize to an unlabeled target. The most famous domain adaptation strategy for image forgery detection is ForensicTransfer [13]. This architecture leverage an autoencoder to learn a latent space separating domain-specific information irrelevant for detection, from domain-invariant information relevant for detection. This approach has shown promising results for robust synthetic image detection when a few labels from the target set are available. Classical strategies from machine learning literature are also used to find feature-invariant spaces with a relevant adaptation cost [14], [15], typically a distance between distributions or an adversarial loss added to the binary cross-entropy loss [16], [17]. While detector-centric strategies are promising, some require labeled target data and others assume a balanced distribution of pristine and manipulated images in their targets. Moreover, models adapted to specific targets tend to overfit and do not generalize well to other targets, requiring practitioners to retrain multiple models for different investigations.

To our knowledge, there is a clear lack of studies on the out-of-distribution robustness of manipulation detectors in scenarios where no target data is available at training time. We highlight this issue by presenting an interesting phenomenon in Table I, which is never mentioned in forensic literature. This table displays the performance of the same architecture trained starting from three different seeds, showing similar performance on the testing set of the source, while exhibiting very different performances on 20 different post-processed targets. We highlight here that different trainings of the same architecture do not lead to the same robustness against post-processing, despite similar performance on the source (convergence towards different local minimums). This disparity raises important questions about the best practices for training to ensure consistent robustness against different post-processing attacks.

B. Contributions

Our primary goal is to understand why the same architecture dedicated to splicing detection can exhibit different behaviors when faced with post-processing attacks. We consider a realistic scenario where a practitioner can train several detectors in various ways and needs to identify the most robust detector for investigations, without targeting a specific post-processing. Our study aims to:

- Explore the factors that make a splicing detector robust to post-processing attacks.
- Derive best practices for forensic practitioners to enhance the performance of their detectors on scrutinized images.

This paper is the first to address the challenge of post-processing domain shift in forensics by proposing multiple trainings of the same architecture. Our contributions are listed as follow:

1) We highlight that highly specific training on a source results in poor generalization performance on post-processed targets.
2) We demonstrate a clear correlation between the separation of training data into pristine and spliced classes within first and last latent spaces and the ability of a detector to generalize to post-processed images.
3) We compare the impact of pooling and normalization layers on the ability of the detector to generalize to a variety of post-processed samples.

The structure of the paper is as follows: Section II presents the formalization of our objective and introduces the notion of latent space margins. In Section III, a series of experiments are conducted to gain a deeper understanding of factors contributing to a robust learning for a splicing detector. The influence of pooling and normalization operators on the robustness of detection is notably examined in this section. Lastly, Section IV serves as the conclusion, summarizing the main findings and contributions of this research and proposing some perspectives.

II. Formalization

A. Problem formulation and scenario

In accordance with [18], we define a processing pipeline as a vector $\omega \in \Omega$ that encompasses all the parameters associated with the pipeline, such as the downsampling factor, the denoising coefficient, the JPEG quality factor, *etc*. For splicing detection, machine learning models are commonly used:

$$f(x \mid \theta_\omega) : \mathcal{X} \to \{pristine, manipulated\}$$
$$x \mapsto y$$

Here, $\theta_\omega \in \Theta$ represents the learned parameters using pristine and spliced images post-processed using parameters ω. To assess the impact of post-processing mismatch, it is common to compute the *generalisation gap* between a source s (training base) and a target t (evaluation base):

$$\mathcal{G}_{f(x|\theta_\omega)}(\omega_s, \omega_t) = \mathbb{E}_{(x,y)\sim P((x,y)|\omega_s)}(f(x \mid \theta_{\omega_s}) = y) - \mathbb{E}_{(x,y)\sim P((x,y)|\omega_t)}(f(x \mid \theta_{\omega_s}) = y) \quad (1).$$

This gap represents the difference of performance between the ideal scenario where the post-processing of the target is the same as the one of the source and the real scenario where we do not know the post-processing of target images.

Here we do not have access to target samples at training time. Ideally, we want to build a detector as robust as possible against unknown post-processings. We assume that a forensic practitioner is familiar with a splicing detector but does not know what to do to make it robust.

B. Latent spaces margins

In splicing detection, we have two classes: *pristine* and *spliced*. Accordingly, our models output two logit scores, f_1 and f_2, for each input $x \in \mathcal{X}$. The class with the highest score is chosen as the predicted label, given by $i^* = \arg\max_i f_i(x)$. Deep detectors are made of successive layers, with each layer

projecting its input into a new latent space[1]. The linear decision boundary in the final latent space appears non-linear in previous latent spaces, leading to a distinct decision boundary for each latent space. The decision boundary $\mathcal{D}^l$ of the l-th latent space of our detector is the set of points in this latent space x^l where the detector is uncertain between the two classes:

$$\mathcal{D}^l = \left\{ x^l \mid f_1(x^l) = f_2(x^l) \right\}. \quad (2)$$

We can now define the margin of a latent sample x^l with respect to this latent boundary $\mathcal{D}^l$ as the smallest perturbation δ^l necessary to move x^l to the decision boundary of the l-th latent space:

$$d^p_{f,x^l} = \min_{\boldsymbol{\delta}} \|\delta^l\|_p \quad \text{s.t.} \quad f_1(x^l + \delta^l) = f_2(x^l + \delta^l) \quad (3).$$

The performance gap caused by post-processing mismatches exists because deep splicing detectors tend to learn biases that are very specific to the training distribution. This creates decision boundaries suited to source samples but ineffective for target samples with different distributions. Ideally, we feel that a general learning would space out samples from different classes more clearly in the latent spaces. Indeed, a boundary too close to source samples means that even slight variations or noise addition in the training data (such as those introduced by post-processing pipelines) can cause the new data from another distribution to overshoot this boundary, leading to misclassifications. A previous research showed a correlation between the generalization gap and the distribution of latent margins (distances between latent decision boundaries and the training points from each class) [19]. However, this study tested this intuition with only two target distributions and used image classification classifiers relying on semantics for their decisions. In this paper, we propose to check the validity of this correlation in the context of splicing detection. We validate this rational using with twenty targets that have undergone various post-processings, using the Bayar detector [1], which bases its decisions on image noise rather than semantics.

III. Best Training Practices for Robust Detection

A. Experimental protocol

1) Detector's choice and hyperparameters: For our experiments, we use the popular forgery detector developed by Bayar and Stamm [1]. This deep detector is known for its simple yet effective design, making it an interesting choice for standard databases and sufficient for our analysis. Its architecture is traditional (Convolution + Max Pooling + Fully Connected Layers). However, the very first convolutional layer is constrained to perform high-pass filtering:

$$\begin{cases} \boldsymbol{w}_k^{(1)}(0,0) = -1, \\ \sum_{m,n\neq 0} \boldsymbol{w}_k^{(1)}(m,n) = 1. \end{cases}$$

This constraint fosters the extraction of relevant low-level forensic features The other layers are non-constrained and act as usual. The following choices were made concerning the optimization strategy and the hyperparameters:

- The maximal number of epochs is fixed at 115, a reasonable amount of epochs enabling to observe a convergence in practice.
- The optimizer is SGD.
- The batch size is fixed at 128, a reasonable size for computation on a regular GPU while ensuring a good convergence of our detector.
- The learning rate (lr) is fixed to 10^{-3} with a lr scheduler dividing by 10 the lr with a patience of 4 epochs.
- The initialization of our weights is the one by default on pytorch [20]. For each study, we initialized our forgery detector with the common seed 22 in order to make our results reproductible and ensuring a fair comparison of them.

We cut source and targets into train/validation/test with the proportion $0.6/0.2/0.2$. To get the best of our model and avoid overfitting, we consider for each experiment an early stopping callback based on the accuracies obtained on the source validation set.

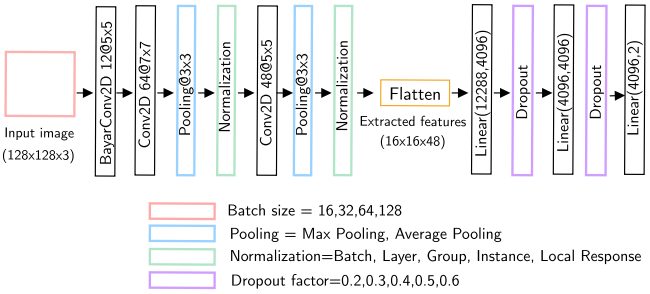

Fig. 1. Scheme of Bayar Detectors [1]. Colored cells indicates the hyperparameters or operators that are changing over our 200 trainings.

2) Construction of source and target domains: The largest public dataset dedicated to splicing detection is DEFACTO [21]. We have chosen this dataset for our analysis due to several key factors: its extensive size, the high-quality and realistic images it contains, and the full control it offers over the post-processing pipelines, with images saved in TIF format. These features make DEFACTO an ideal choice for our study.

To prepare source and targets datasets for our experiments, we start by dividing the splicing category of DEFACTO into two equally-sized, independent sets: one for the source and one for targets. Each image is then cut into 128×128 patches to ensure a uniform training process. For each patch from both source and target bases, we create a *spliced* class selecting patches with a tampered surface ratio between 10% and 40% of their total area. This specific range is carefully chosen to balance two factors: if the tampered area is too small or too high, the detector may struggle to differentiate two noise distributions. By selecting carefully our patches, we aim to create a realistic and challenging environment for evaluating the performance of our splicing detectors. Based on the number of spliced patches, *pristine* patches are then

[1]A latent space is a lower-dimensional representation of data capturing its underlying structure and features.

selected in equal quantity to constitutes balanced classes. This preprocessing gives us around 20.000 patches for our training sets and 7.000 patches for our testing sets. To build our source, we only work with original TIF images of DEFACTO. For the targets, post-processing are applied. All these processing are done on the original TIF images before cutting them into patches to prevent artifacts. Note that patches from two distinct post-processed targets are similar in terms of content. This is done to uniquely attribute the observed generalization gap to the application of the post-processing pipeline and not to the content associated to the training or testing sets.

3) Post-Processing Pipelines: We created a set of 20 processing pipelines playing with Wavelet Denoising and Sharpening operations of RawTherapee, an open source software for image processing. We apply JPEG compression with a quality factor of 70 at the end of each pipeline using Imagemagick to fully control it. Our choice of pipelines was a good tradeoff between target realism and the observation of strong generalization gaps. Details about these post-processing pipelines are presented in Figure 2.

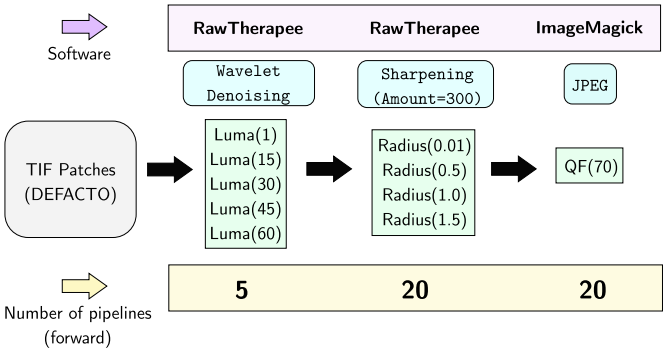

Fig. 2. Details about the post-processing pipelines applied to the target TIF set for our study.

4) Multiple trainings of the same architecture: We propose here to play with hyperparameters and operators of the Bayar detector to study how they influence its ability to generalize to post-processed samples. We perform 200 trainings of the Bayar Detector modifying batch size, pooling, normalization and drop out following the scheme of Figure 1.

5) Quantile plots for analysis: Quantile plots help us visualize how the generalization gap distribution evolves within a sliding window centered in successive values of diverse metrics. For each quantile plot, we scan metric points with a step of 0.01 and a window size of 0.2, balancing the tradeoff between localization and accuracy on quantiles computation.

B. Source overfitting through the epochs

The results obtained from our 200 trainings validate the issue of source overfitting briefly mentioned in [19]: as the accuracy on the source test set increases, the generalization gap across all target domains also increases. We believe this happens because the network learn more and more source-specific features over the epochs, ultimately focusing on specific biases present in the source samples to enhance its performance on this source. This observation is illustrated in Figure 3. It shows how the generalization gap evolves w.r.t. the final accuracy on the source. Therefore, while forensic analysts should strive for good performance on their source, we do not recommend excessive trainings of splicing detectors if they have to evaluate their detectors on unknown targets.

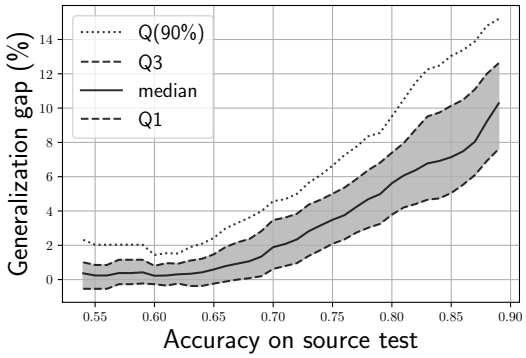

Fig. 3. Quantile plots representing the evolution of the generalization gap over our 20 domains according to the accuracy of our detectors on the source test. Q1 is the first quartile, Q3 is the third quartile and Q(90%) is the 90th percentile.

C. Latent margins and generalization gaps

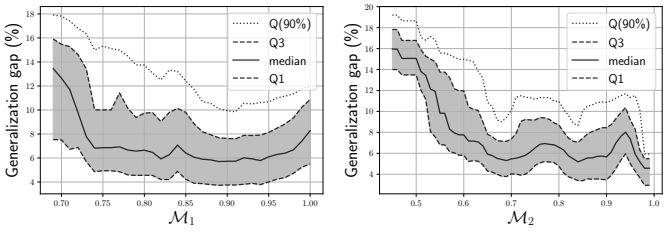

Fig. 4. Quantile plots representing the evolution of the generalization gap over our 20 domains according to the margin metrics $\mathcal{M}_1$ and $\mathcal{M}_2$ computed using latent margins from all layers. These metrics are normalized for comparison. Q1 is the first quartile, Q3 is the third quartile and Q(90%) is the 90th percentile.

Here we want to check if there is a correlation between distances to the boundary of well-classified latent samples and generalization gaps. Given that all our models did not converge equally, we restrict our studies to the 138 models achieving an accuracy of at least 75% on the source, in order to discard low generalization gaps caused by underfitting. For the computation of latent margins, we follow a methodology inspired by [19]:

1) Estimate the latent margins d^2_{f,x^l} of each source sample x^l using logits f_1, f_2 and their gradients w.r.t. each layer, while taking care of normalizing them for scale independence. More details about this computation are available in [19] and our github repo. Following [19], we discard negative margins caused by misclassifications.
2) Summarize margins distributions per latent space with vectors $\boldsymbol{\mu_l}$ of descriptive statistics (first and third quartiles, median, upper and lower fences). Eventually, one could mix all the $\boldsymbol{\mu_l}$ in one vector $\boldsymbol{\mu}$.

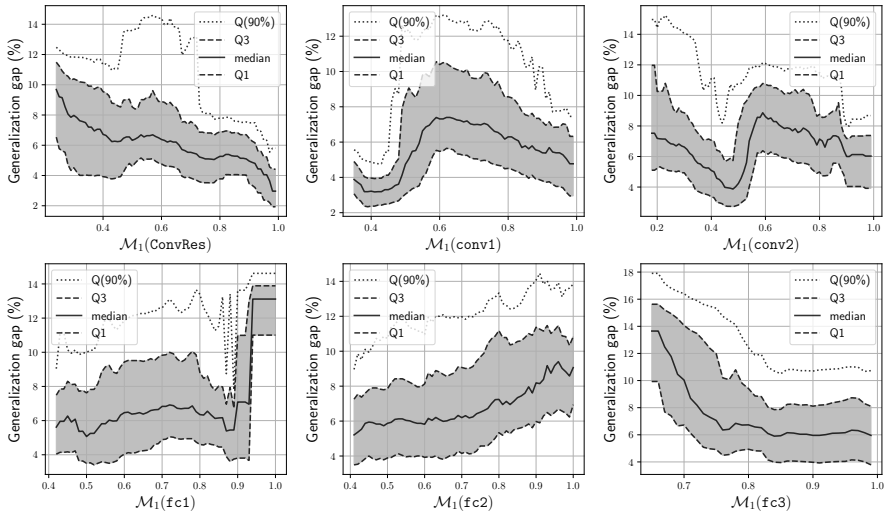

Fig. 5. Quantile plots representing the evolution of the generalization gap over our 20 domains according to $\mathcal{M}_1$ computed with latent margins relative to a single layer of our Bayar detectors. These metrics are normalized for comparison. Q1 is the first quartile, Q3 is the third quartile and Q(90%) is the 90th percentile.

3) Combine the margin statistics of every vector to derive a margin metric $\mathcal{M}$. We suggest testing straightforward combinations by computing the sum of the statistics raised to a certain power α : $\mathcal{M}_\alpha = \sum_i \mu_i^\alpha$. Raising α enable to better emphasize margin differences among the architecture.

Contrary to [19], our goal is not to predict the generalization gap with margin statistics but rather provide practitioners a metric enabling to assess the quality of their training for robust splicing detection. For each Bayar Detector $f^i(x|\theta_s)$ trained on our source s and each target t post-processed with pipeline ω_t, we produce couples $[\mathcal{M}_\alpha[f^i(x|\theta_s)]\ ,\ \mathcal{G}_{f^i(x|\theta_{\omega_s})}(\omega_s,\omega_t)]$. Such couples disclose the eventual presence of a correlation between the $\mathcal{M}_\alpha$ and the generalization gaps over 2760 points. Quantile plots of Figure 4 confirm that most robust detectors against post-processing attacks are the ones separating the most their latent training samples (especially with the use of $\mathcal{M}_2$).

D. Importance of the margin in each latent space

Until here, we used the margin distributions of each latent space of our Bayar detectors for the computation of $\mathcal{M}_\alpha$.

Although [19] argues that examining the margins of a single latent space is inadequate for capturing the generalization gap, we still propose to investigate the correlation between $\mathcal{G}$ and $\mathcal{M}_1$ computing the margin statistics from each layer independently through Figure 5. This choice is guided by the observation of a small positive correlation between G and $\mathcal{M}_1$ when we reach high margins that we would like to understand. We see that the expected correlation between $\mathcal{G}$ and $\mathcal{M}_1$ is clearly present using margin distributions from the very first and the very last latent spaces. However, this correlation does not hold for the intermediate latent spaces. We explain the correlation with the very first layer by the fact that upstreams layers are known to extract more general features than downstream layers in deep architectures [22]. Hence, maximizing source margins at that level is also expected to maximize target margins at this same level, shrinking the final generalisation gap. Concerning the very last layer, its the most specific layer of the architecture. Hence, if source margins are too tight at that level, noisy perturbation of the source (i.e. post-processings) lead to misclassifications. We believe this explains why a large class separation in this latent space is also beneficial for robustness against post-processing. This justify the recent trend towards the construction of a final latent space clearly separating the classes using contrastive losses [23]. Regarding the intermediate latent spaces, the absence of negative correlation between $\mathcal{M}_1$ and $\mathcal{G}$ is certainly because these layers are trained to separate classes in the final latent space without bothering to well separate them within their own latent space.

Operator name	Median Source Accuracy	Median $\mathcal{M}_2$	Min $\mathcal{G}$	Q1 $\mathcal{G}$	Median $\mathcal{G}$	Q3 $\mathcal{G}$	Max $\mathcal{G}$
Normalization							
instancenorm	80%	0.83	0%	2%	3%	4%	6%
batchnorm	82%	0.66	1%	7%	10%	13%	21%
layernorm	82%	0.82	0%	4%	6%	7%	11%
local_response_norm	83%	0.63	0%	4%	6%	7%	16%
group	85%	0.81	0%	5%	7%	9%	15%
Pooling							
average_pooling	82%	0.83	0%	4%	7%	10%	15%
max_pooling	83%	0.69	0%	5%	6%	8%	21%
Dropout							
dropout(0.2)	82%	0.77	0%	4%	6%	7%	15%
dropout(0.3)	83%	0.76	0%	4%	6%	9%	16%
dropout(0.4)	82%	0.77	0%	4%	6%	8%	16%
dropout(0.5)	83%	0.76	0%	4%	7%	10%	20%
dropout(0.6)	83%	0.79	0%	5%	8%	11%	21%
BatchSize							
16	84%	0.79	1%	5%	7%	10%	20%
32	83%	0.78	1%	5%	7%	10%	18%
64	80%	0.76	0%	4%	6%	8%	21%
128	77%	0.69	0%	3%	5%	6%	11%

TABLE II

CONTRIBUTION OF NORMALIZATION, POOLING, DROPOUT, AND BATCH SIZE. $\mathcal{M}_2$ IS NORMALIZED SIMILARLY AS IN FIGURE 4. THE LOWEST SOURCE ACCURACY, THE HIGHEST MARGINS AND THE LOWEST GENERALIZATION GAPS ARE HIGHLIGHTED IN GREEN.

E. Impact of parameters and operators on robust detection

Here we propose to examine the impact of hyperparameters and operators on generalization gaps and the distributions of latent margins. For this analysis, we compute $\mathcal{M}_2$ using only the margins from first and last layers to enhance the correlation of our margin metric with the generalization gap.

We provide statistics helping to assess the impact of independently modifying hyperparameters and operators on the generalization ability of our detectors in Table II. Our results demonstrate that `instance normalization` and `average pooling` are particularly effective choices for designing robust splicing detectors. We attribute this effectiveness through high median $\mathcal{M}_2$, showing that these operations significantly expand latent margins. Regarding hyperparameters, the most robust architectures are, as expected, those with the lowest median accuracy on the source. However, we must acknowledge that modifying `dropout` does not have a significant impact on robust detection, while modifying `batch size` has an impact that is not really explained by high margins. We believe this is because dropout and batch size are hyperparameters that have less impact on latent representations compared to normalization and pooling operators that respectively squeeze data and reduce its dimensionality.

IV. Conclusions and Perspectives

This article explores how the robustness of a splicing detector against unknown post-processing can vary depending on its training. By examining factors that influence detector performance, we proposed several best practices for forensic analysts. Our research first showed that over-training a detector on a single source negatively affects its generalization to post-processed samples, prompting the need to determine optimal training stopping points. To help with this, we developed a margin metric correlated with the generalization gap by leveraging classical statistics to summarize latent margins distributions. Notably, we found that the first and last latent margins distributions significantly correlate with the detector's robustness against post-processing. Finally, we discovered that some pooling and normalization operators proved more effective than others in fostering post-processing robustness given the wide latent margins they produced. We currently recommend training splicing detectors with multiple hyperparameters choice and selecting the one maximizing margins in first and last layers. In future research, we plan to check the consistence of this correlation with different forgery detectors such as Noiseprint [24] and Trufor [3]. If the correlation persists, we propose to design architectures resilient to post-processing robustness by maximizing latent margins, leveraging for instance the contrastive losses that already proved effective in forensics [23]. We also intend to study the effects of different hyperparameter and operator combinations on out-of-distribution (OOD) generalization for splicing detectors.

References

[1] B. Bayar and M. C. Stamm, "A deep learning approach to universal image manipulation detection using a new convolutional layer," ser. IH&MMSec '16. New York, NY, USA: Association for Computing Machinery, 2016, p. 5–10.

[2] M.-J. Kwon, S.-H. Nam, I.-J. Yu, H.-K. Lee, and C. Kim, "Learning jpeg compression artifacts for image manipulation detection and localization," *International Journal of Computer Vision*, vol. 130, no. 8, pp. 1875–1895, Aug. 2022.

[3] F. Guillaro, D. Cozzolino, A. Sud, N. Dufour, and L. Verdoliva, "Trufor: Leveraging all-round clues for trustworthy image forgery detection and localization," 2023.

[4] D. Cozzolino and L. Verdoliva, "Single-image splicing localization through autoencoder-based anomaly detection," in *2016 IEEE International Workshop on Information Forensics and Security (WIFS)*, 2016, pp. 1–6.

[5] F. Marra, D. Gragnaniello, L. Verdoliva, and G. Poggi, "A full-image full-resolution end-to-end-trainable CNN framework for image forgery detection," *CoRR*, vol. abs/1909.06751, 2019.

[6] X. Luo, R. Zhan, H. Chang, F. Yang, and P. Milanfar, "Distortion agnostic deep watermarking," *CoRR*, vol. abs/2001.04580, 2020.

[7] E. Giboulot, R. Cogranne, D. Borghys, and P. Bas, "Effects and Solutions of Cover-Source Mismatch in Image Steganalysis," *Signal Processing: Image Communication*, Aug. 2020.

[8] B. Ahmed, T. A. Gulliver, and S. alZahir, "Image splicing detection using mask-rcnn," *Signal, Image and Video Processing*, vol. 14, pp. 1035–1042, 2020.

[9] X. Xu, J. Dong, W. Wang, and T. Tan, "Robust steganalysis based on training set construction and ensemble classifiers weighting," in *2015 IEEE International Conference on Image Processing (ICIP)*, 2015, pp. 1498–1502.

[10] R. Abecidan, V. Itier, J. Boulanger, P. Bas, and T. Pevný, "Using Set Covering to Generate Databases for Holistic Steganalysis," in *IEEE International Workshop on Information Forensics and Security (WIFS 2022)*, Shanghai, China, Dec. 2022.

[11] T. Pevny and J. Fridrich, "Detection of double-compression in jpeg images for applications in steganography," *IEEE Transactions on Information Forensics and Security*, vol. 3, no. 2, pp. 247–258, 2008.

[12] R. Abecidan, V. Itier, J. Boulanger, P. Bas, and T. Pevný, "Leveraging Data Geometry to Mitigate CSM in Steganalysis," in *IEEE International Workshop on Information Forensics and Security (WIFS 2023)*, Nuremberg, Germany, Dec. 2023.

[13] D. Cozzolino, J. Thies, A. Rössler, C. Riess, M. Nießner, and L. Verdoliva, "Forensictransfer: Weakly-supervised domain adaptation for forgery detection," *CoRR*, vol. abs/1812.02510, 2018. [Online]. Available: http://arxiv.org/abs/1812.02510

[14] Y. Ganin and V. Lempitsky, "Unsupervised domain adaptation by backpropagation," in *International conference on machine learning*. PMLR, 2015, pp. 1180–1189.

[15] M. Long, Y. Cao, J. Wang, and M. Jordan, "Learning transferable features with deep adaptation networks," in *International conference on machine learning*. PMLR, 2015, pp. 97–105.

[16] A. Kumar and A. Bhavasar, "Syn2real: Forgery classification via unsupervised domain adaptation," *CoRR*, vol. abs/2002.00807, 2020.

[17] R. Abecidan, V. Itier, J. Boulanger, and P. Bas, "Unsupervised JPEG Domain Adaptation for Practical Digital Image Forensics," in *IEEE International Workshop on Information Forensics and Security (WIFS 2021)*, Montpellier, France, Dec. 2021.

[18] D. Šepák, L. Adam, and T. Pevný, "Formalizing cover-source mismatch as a robust optimization," in *EUSIPCO: European Signal Processing Conference*, Belgrade, Serbia, Sep. 2022.

[19] Y. Jiang, D. Krishnan, H. Mobahi, and S. Bengio, "Predicting the generalization gap in deep networks with margin distributions," 2019.

[20] K. He, X. Zhang, S. Ren, and J. Sun, "Delving deep into rectifiers: Surpassing human-level performance on imagenet classification," in *Proceedings of the IEEE international conference on computer vision*, 2015, pp. 1026–1034.

[21] G. Mahfoudi, B. Tajini, F. Retraint, F. Morain-Nicolier, J. L. Dugelay, and M. Pic, "DEFACTO: image and face manipulation dataset," in *27th European Signal Processing Conference (EUSIPCO 2019)*, 2019.

[22] J. Yosinski, J. Clune, Y. Bengio, and H. Lipson, "How transferable are features in deep neural networks?" *CoRR*, vol. abs/1411.1792, 2014.

[23] D. Cozzolino, D. Gragnaniello, G. Poggi, and L. Verdoliva, "Towards universal gan image detection," 2021.

[24] D. Cozzolino and L. Verdoliva, "Noiseprint: a cnn-based camera model fingerprint," *CoRR*, vol. abs/1808.08396, 2018. [Online]. Available: http://arxiv.org/abs/1808.08396