Efficient Anomaly Detection Using Self-Supervised Multi-Cue Tasks

Loïc Jézéquel, Graduate Student Member, IEEE, Ngoc-Son Vu, Jean Beaudet, and Aymeric Histace

Abstract—Anomaly detection is important in many real-life applications. Recently, self-supervised learning has greatly helped deep anomaly detection by recognizing several geometric transformations. However, these methods lack finer features, usually highly dependent on the anomaly type, and do not perform well on fine-grained problems. To address these issues, we first introduce in this work three novel and efficient discriminative and generative tasks which have complementary strength: (i) a piece-wise jigsaw puzzle task focuses on structure cues; (ii) a tint rotation recognition is used within each piece, taking into account the colorimetry information; (iii) and a partial re-colorization task considers the image texture. In order to make the re-colorization task more object-oriented than background-oriented, we propose to include the contextual color information of the image border via an attention mechanism. We then present a new out-of-distribution detection function and highlight its better stability compared to existing methods. Along with it, we also experiment different score fusion functions. Finally, we evaluate our method on an extensive protocol composed of various anomaly types, from object anomalies, style anomalies with fine-grained classification to local anomalies with face anti-spoofing datasets. Our model significantly outperforms state-of-the-art with up to 36% relative error improvement on object anomalies and 40% on face anti-spoofing problems.

Index Terms—Anomaly detection, fine grained classification, self-supervised learning, multi-task learning, one-class learning.

I. INTRODUCTION

ONE of the most fundamental challenge in machine learning is detecting an observation as anomalous compared to a normal baseline. Properly solving such problem with high predictability and robustness has been essential in many fields. To mention a few, in intrusion detection [1] where we wish to detect untrustworthy entries on a network, fraud detection [2] where a forged item or transaction must be rejected, in medical imaging [3] where abnormalities in a captured image must be located, video surveillance [4], [5] where abnormal events are detected, and in manufacturing defect detection [6], [7].

With the advent of deep learning, many tasks on image data including binary classification and anomaly detection (AD) have greatly improved. Nevertheless classical binary classification still generally lacks robustness and reliability outside its training domain. Many anomaly detection methods try to solve this problem by only learning the normal class boundary, rather than directly discriminating anomalies from normal samples. Any observation defined outside is then deemed as anomalous. This decision rule is especially useful when the anomaly class boundary is ill-defined or continually evolving and only few anomalous training samples are available.

The recent explosion of self supervision further improves unsupervised learning abilities and reduces the needed amount of labeled data. It enables to discriminate anomalies from normal samples by learning to solve simple tasks such as geometric transformation classification. However, although deep anomaly detection can achieve interesting performance, it still suffers from limitations on more challenging problems with local and fine-grained differences between anomalies and normal samples. Indeed, existing self-supervised anomaly detection algorithms evaluated their performance on datasets like CIFAR10 or CIFAR100 but not on fine-grained ones like Caltech-Birds or face anti-spoofing. Moreover, these methods usually have an high inference time, making them impractical for real-life anomaly detection problems. For example, the state-of-the-art model GeoTrans [8] needs to apply during inference 72 different transformations to the input making it around 10 times slower than our proposed method.

In this given context, our main contributions in this paper are the following:

- We introduce a new way to efficiently exploit the benefits of discriminative and generative auxiliary tasks in self-supervised anomaly detection. Using the two-branch network, we are among the first to reach high-quality results with auxiliary tasks on fine-grained anomaly detection and face anti-spoofing in a one-class setting.
- We carefully design and optimize three novel specialized auxiliary tasks according to loss functions, anomaly scores as well as complexity. This allows our model to learn very rich and complementary representations which better encompass image structure (Section III-A), colorimetry (Section III-B) and texture (Section III-D).
- With these tasks, we also explore different out-of-distribution (OOD) detection methods and fusion functions.

Manuscript received 2 May 2022; revised 12 October 2022; accepted 2 December 2022. Date of publication 9 January 2023; date of current version 13 January 2023. The associate editor coordinating the review of this manuscript and approving it for publication was Dr. Zhao Zhang. (Corresponding author: Loïc Jézéquel.)

Loïc Jézéquel is with ETIS UMR 8051 (CY Cergy Paris Université, ENSEA, CNRS), 95000 Paris, France, and also with the Idemia Identity and Security, 95520 Osny, France (e-mail: loic.jezequel@ensea.fr).

Ngoc-Son Vu and Aymeric Histace are with ETIS UMR 8051 (CY Cergy Paris Université, ENSEA, CNRS), 95000 Paris, France (e-mail: son.vu@ensea.fr, aymeric.histace@ensea.fr).

Jean Beaudet is with the Idemia Identity and Security, 95520 Osny, France.

Digital Object Identifier 10.1109/TIP.2022.3231532

1941-0042 © 2023 IEEE. Personal use is permitted, but republication/redistribution requires IEEE permission. See https://www.ieee.org/publications/rights/index.html for more information.
• We compare our method with state-of-the-art using an exhaustive protocol for anomaly detection covering object, style and local anomalies, and even more challenging task of face anti-spoofing.
• The proposed method obtains high-quality results with up to 36% AUROC relative improvement on object anomalies and 53% on face anti-spoofing from state-of-the-art anomaly detection methods.

This paper follows the motivation of our work presented in [9]. In [9], we improved the anomaly detection by simultaneously solving in a self-supervised fashion a high-scale geometric task and a low-scale jigsaw puzzle task. It is worth noting that the differences of this paper compared to [9] are significant: all pretext tasks are novel and more efficient. In this paper, we address the inference complexity issue and considerably improve the anomaly detection performance.

First, we give an overview of anomaly detection related work in Section II. Then we present our new pretext tasks in Section III, and our study of OOD methods with fusion in Section IV. Our complete model is summarized in Section V which we give a general overview in Fig. 3. In a first stage, a jigsaw puzzle task with intra-piece tint rotation detection and a partial colorization are performed. Then in a second stage, a set of OOD scores is computed for each task and is aggregated into a single anomaly score using a fusion function. In addition, we extensively compare our model with state-of-the-art in Section VI, and provide several experiments on the influence of our model parameters in Section VII. Finally, we discuss future work in Section VIII.

II. RELATED WORK

We first review several common classical and deep anomaly detection methods in Section IIA and Section IIB. We then present self-supervised learning and how they are applied for AD in Section IIC and Section IID, respectively. Readers are referred to [10], [11], and [12] for more in-depth surveys on AD or self-supervised learning.

A. Classical Anomaly Detection

The main goal in anomaly detection is to classify a sample as normal or anomalous. Formally, we predict \( P(x \in X_{\text{norm}}) \) for an observation \( x \) and a normal (or positive) class \( X_{\text{norm}} \). The anomalous (or negative) class is then defined implicitly as the complementary of the normal class in image space. We can generally categorize anomalies into three families:
1) **Object anomaly**: any object which is not included in the positive class, e.g., a cat is an object anomaly in regards to dogs.
2) **Style anomaly**: observations representing the same object as the positive class but with a different style or support, e.g., a realistic mask or a printed face represent faces but with a visible different style.
3) **Local anomaly**: observations representing and sharing the same style as the positive class, however a localized part of the image is different. Most of the time, these anomalies are the superposition of two generative processes, e.g., a fake nose on a real face is a local anomaly.

Usually, we assume in anomaly detection that only normal samples are available during training, meaning that methods are in one-class setting. Traditionally, one-class Support Vector Machine [13] (OC-SVM) or its extension the Support Vector Data Description [14] (SVDD) were used for anomaly detection. The anomaly score of an observation \( x \) is given by its distance to a parameterized boundary \( \Omega \). OC-SVM defines \( \Omega \) as an hyper-plan separating the origin from the normal samples with the maximum margin, whereas SVDD uses an hyper-sphere containing all normal samples with the minimum radius (see Fig. 1(a,b)).

Fully-unsupervised methods which learn from a set of unlabeled data containing normal samples and anomalies were also used. Such non-deep methods include Robust Principal Component Analysis [15] (RPCA) or the Isolation Forest (IF) [16]. Rather than modeling the normal samples, the IF algorithm tries to isolate anomalies from normal samples via successive random partitions of the feature space. If the sample can be entirely isolated (i.e. be the only point in a region) in a few partitions, then it is more likely to be anomalous (see Fig. 1(c)).

These classical methods have shown great success on low-dimensional data such as tabular data, but usually fail on higher dimension inputs such as images.

B. Deep Anomaly Detection

The introduction of neural networks as feature extractors gave birth to several hybrid methods where a pre-trained neural network is used to extract features, on which a classical algorithm such as OC-SVM or isolation forest is trained. It ultimately led to the first end-to-end anomaly detection neural network, the one-class Neural Network (OC-NN) [17] which integrates the OC-SVM loss in the network training. More recent methods include different dedicated approach to anomaly detection. In [18], [19], and [20], a binary classification is used with pseudo negative images or latent vectors to represent the anomaly class. Another approach is to use the error of a generative model reconstruction [21], [22], [23], [24] or the gradient of the error given that the image is normal [25]. Finally, the self-supervision framework can be used to learn normal class representations and subsequently form an anomaly score as presented in Section IID.
There also have been semi-supervised anomaly detection methods such as DeepSAD [26] or deviation networks [27] where we assume some of the anomalies representing a few modes are available. These methods can achieve better accuracy on borderline cases given enough diverse anomalies, which is often less manageable in practice. In particular, these two methods directly learn representations by minimizing the distance of normal sample features to an hypersphere center, while maximizing the distance to the anomalies. It follows the compactness principle, where the normal class representations variance is minimized and the inter-class representations variance is maximized.

C. Self-Supervised Learning

Self supervised learning (SSL) is a part of representation learning, where useful and general representations are learned from an unlabeled dataset. The learned features are then used through transfer learning for a different task such as classification.

In this manner, representations are learned by solving from the data an auxiliary task \( T \), which is often unrelated to the final one. The pretext task can either be discriminative, usually resulting in a multi-class classification setting or generative where a regression loss is often utilized. Any SSL is defined by its pretext objective loss \( L \) and its pretext data generation function \( DG_T : \mathcal{P}(X) \mapsto \mathcal{P}(X \times K) \) which yields a labeled set from an unlabeled set \( X \). In the case of discriminative tasks, it is usually done via \( n \) images transformations \( T_1, \cdots, T_n \):

\[
DG_T (\{x_i\}_{i \in \{1..N\}}) = \{(T_j(x_i), j)\}_{i \in \{1..N\}, j \in \{1..n\}}
\]  

(1)

where the \( x_i \) are images from the unlabeled training dataset.

In other words, SSL consists of two steps: (1) generating a labeled set \( X_T = DG_T (X) \), (2) training a classification or regression network on this generated labeled set. One of the final layers are then used as a feature extractor. Some commonly used tasks are: 90° rotation prediction [28], jigsaw puzzle [29], distortions [30], colorization [31], image inpainting [32] or relative patches prediction [33].

More recently, the contrastive learning framework [34] has been extensively used for self-supervised representation learning. Unlike the methods above, it does not rely on an explicit pretext task and directly formulates losses on the representations. The most effective contrastive method is instance discrimination [35], [36] where the objective is to maximize similarity between augmented versions of a same image (positive samples) while minimizing similarity with any other images (negative samples). The instance discrimination can be seen as a pretext task where the pretext data generation function maps samples to the set of positive pairs and negative pairs and the objective function is to discriminate positive from negative pairs using cosine similarity in representation space.

D. SSL Anomaly Detection

In this section, we first present how to apply SSL for AD and then discuss some state-of-the-art methods exploiting SSL for AD.

Very recently, SSL has been adapted to the one-class anomaly detection framework. First we learn to solve an auxiliary task in a SSL fashion. Then, a measure of how well the network can solve the task on the generated dataset \( DG_T (X) \) is used to classify at inference time an observation \( x \) as anomalous or normal. The main assumption is that the network will perform relatively well on normal samples but will fail on anomalies. The goals of representation learning and AD are different. In representation learning we try to maximize the performance of the representation on as many downstream tasks and data as possible; whereas in AD, we want a clear discrimination through performance on normal and anomalous data.

Any SSL anomaly detector is composed of three steps (see Fig. 2):

1) The representation learning on the normal class, carried out in a self-supervised manner. In our case this is done by solving a pretext task \( T \), but other methods employ other mechanisms such as contrastive learning.

2) During inference of an unseen sample \( x \), an out-of-distribution (OOD) detection method is applied on the generated labeled samples \( DG_T (\{x\}) \). The goal of OOD methods is to detect whether or not an observation has been sampled from the same distribution as the training set. OOD is more low-level and general than AD, and aims at modeling the training distribution rather than the normal class. For example, contrary to AD the CIFAR-100 dataset would be considered out of distribution in regards to CIFAR-10. Given a pre-trained model \( \Psi \) on a distribution \( F_{\text{train}} \), it estimates \( P(x \sim F_{\text{train}}) \). The normal training set is assumed to be close enough to the real distribution of normal samples, and since we have access to the correct task label \( y \), the following approximations hold:

\[
s_{ood}((x, y); \Psi) \approx P(x \sim F_{\text{train}}) \approx P(x \in \mathcal{X}_{\text{norm}})
\]

(2)

where \( s_{ood}((x, y); \Psi) \) is the OOD score for an image \( x \) with its label \( y \) given the pre-trained network \( \Psi \).

3) The fusion of the OOD scores into a single anomaly score \( s_a \) using a fusion function \( M \).

In the rest of this section, we detail several state-of-the-art self-supervised anomaly detection algorithms that are the most closely related to our work.

In GeoTrans [8], the auxiliary task is to classify which geometrical transformation has been applied to the input from a set \( \{T_i\} \) of 72 random composition of translations, rotations and symmetries. At the end of training, a Dirichlet distribution parameterized by \( \tilde{a}_i \) is fitted over the softmax responses of each transformation on the normal class \( y (T_i(x)) = \max (\phi \circ f(x)) \); then its log-likelihood is used during inference.

\[
s_a(x) = \sum_{i=1}^{72} (\tilde{a}_i - 1) \cdot \log y (T_i(x))
\]

(3)

In MHROT [37], the task is to simultaneously classify 90° rotations, horizontal translations (VTrans), and vertical translation (HTrans), each modeled by a softmax head.
Accordingly, the pretext data generation function is the composition $T_{r,s,t} = \text{Rot}(r) \circ \text{HTrans}(s) \circ \text{VTrans}(t)$, where $r \in \{0^\circ, 90^\circ, 180^\circ, 270^\circ\}$, $s \in \{0, -t_y, +t_y\}$ and $t \in \{0, -t_x, +t_x\}$. During inference, the three softmax of the known transformations for each of the 36 transformation compositions are summed as anomaly score:

$$s_a(x) = \sum_r \sum_s \sum_t y(T_{r,s,t}(x))$$  \hspace{1cm} (4)

Another class of models, called two-stage anomaly detectors [38], does not use the representation learning task during inference, but rather directly apply OOD methods on the representation space. For example, in SSD [40] the representation learning step is performed through contrastive learning, then OOD detection is applied on the representation space induced by the encoder $\phi$. The training data representations are clustered around several centroids using K-means. The Mahalanobis distance is used to compute the anomaly score:

$$s_y(x) = \min_{m} (\phi(x) - \mu_m)^T \Sigma_m^{-1} (\phi(x) - \mu_m)$$  \hspace{1cm} (5)

Similarly, DROC-contrastive (Deep Representation One-class Classification) [38] first learn self-supervised representations from one-class data, and then build one-class classifiers on learned representations. Contrastive learning with distribution augmentation is used for the self-supervised representation learning, and a OC-SVM for the one-class classification.

Finally, it is interesting noting that some SSL anomaly detectors solve the more specific task of anomaly segmentation like CutPaste [43], SOMAD [44]. Those anomaly segmentation consists in predicting a heatmap where the anomaly score is computed on each pixels of the input image. They usually consider very minute and local AD, such as defect detection, while in this work we focus on image-level anomaly detection.

### III. Novel Pretext Tasks

In the rest of the paper, we consider an observation $z$, its label $y$ and a pre-trained network $\phi \circ f$. We gradually detail the proposed pretext tasks for anomaly detection which focus on different visual cues: structure, colorimetry and texture. The tasks of piece-wise puzzle, tint rotation and their combination are discriminative (Sections III-A, III-B, III-C) whereas the colorization task is generative (Section III-D). An overview of the loss function and anomaly score for each proposed task is shown in Table I.

#### A. Piece-Wise Puzzle Task

The puzzle task has been successfully used as a pretext task for representation learning [29], [45]. First an image is separated into $n = n_w \times n_h$ pieces, with some random margin between them. Then given the an image generated by shuffling pieces, a deep encoder is trained to predict which permutation has been applied. It is therefore formulated as a classification task where the prediction label corresponds to the index of the permutation among the $n!$ total possibilities. When the number of pieces becomes too large, the full task is not conceivable and the model should only learn to classify a smaller random subset of all permutations. This formulation of the jigsaw puzzle task, used in our previous work [9] along with geometrical transformation recognition, enables our model to learn low-scale fine features. In the rest of the paper, we call this formulation the partial puzzle task. It is worth noting that regarding to our previous work [9], this paper reconsiders only the puzzle task which is further optimized in both term of time and performance, as will be described in the rest of this section, while other tasks including tint rotation and partial colorization have never been used for visual anomaly detection in the literature, to the best of our knowledge.

The partial puzzle task [9] has several limitations: (i) the quality of the representation highly depends on the chosen permutations. Indeed if the sampled permutations are too hard (e.g. swapping two corners) or too easy, the learned representations will suffer; (ii) Moreover from an anomaly detection perspective, all mispredicted permutations are equally penalized regardless of the number of misplaced pieces.

To address these limitations, we propose here an improved piece-wise puzzle task. Rather than predicting the permutation index, we train a deep encoder to predict the original position of each piece. By assuming each piece is independent, we can now cover all the permutations with only $n^2$ outputs instead of $n!$. Thereby we separate the output layer $f$ into $n$ functions $f_1, \cdots, f_n$, each corresponding to a piece.

Let $\Pi$ be a random permutation, $\Pi(I)$ corresponds to the image $I$ where each piece has been moved according to $\Pi$, and $\Pi_i$ corresponds to the new position of the $i^{th}$ piece. The task is learned using the cross-entropy loss $\mathcal{L}_{CE}$ on every piece predictions:

$$\mathcal{L}_{pul}(\Pi(I)) = \frac{1}{n} \sum_{i=1}^{n} \mathcal{L}_{CE}(\phi \circ f_i(\Pi(I)); \Pi_i)$$  \hspace{1cm} (6)

The full task is illustrated in Fig. 4. In practice, we sample during every training epoch a random subset of $n_{top}$ permutations for each normal image. In order to have as many different permutations as possible in the training set, we define
Fig. 3. Method overview. Our model consists of discriminative (upper U-branch) and generative (lower L-branch) tasks. All the discriminative tasks share the same encoder.

TABLE I
OVERVIEW OF THE LOSS FUNCTION AND OOD SCORE FOR EACH PROPOSED TASK. UPPER U-BRANCH CONSISTS OF PIECE-WISE PUZZLE, TINT ROTATION TASKS WHILE LOWER L-BRANCH CONSISTS OF PARTIAL COLORIZATION TASK

| Task (type)                  | Loss                                                                 | Anomaly score  |
|------------------------------|----------------------------------------------------------------------|-----------------|
| Piece-wise puzzle            | \( L_{\text{ps}}(I) \propto \sum_{i=1}^{n} L_{\text{CE}}(\phi \circ f_1(I); \Pi_i) \) | \( s(I) = \frac{1}{n} \sum_{i=1}^{n} s_{\text{OOD}}((\Pi(I), \Pi_i), \phi \circ f_1) \) |
| Tint rotation                | \( L_{\text{int}}(I) \propto \mathbb{E}_{\phi \sim \Phi} [\| I - \gamma(I, \theta - \Theta) \|_1] \) | \( s(\gamma(I, \theta)) = \sum_{k=1}^{c} \max(\phi \circ f(\gamma(I, \theta)))_{k} \left( \frac{\| I - \gamma(I, \theta + \frac{2\pi k}{c}) \|_1}{\| I - \gamma(I, \theta) \|_1} \right) \) |
| Partial colorization         | \( L_{\text{col}}(I) = \sum_{i,j} \sum_{k=1}^{c} Q_{\text{RM}}(n_{ij}^{(k)}, \mu_{ij}^{(k)}, \Sigma_{ij}^{(k)}) \) | \( s(A_{ij}, B_{ij}|\Pi_{\text{part}}) = \sum_{k=1}^{c} \pi_{ij}^{(k)} N(\| A_{ij}, B_{ij}; m_{ij}^{(k)}, c_{ij}^{(k)} \|) \) |

where \( n_{\text{sp}} = \frac{n!}{N_{\text{train}} \cdot \text{ep}} \), where \( N_{\text{train}} \) is the size of the training set and \( \text{ep} \) the number of training epochs.

During inference we also consider a random subset of \( n_{\text{sp}} \) permutations, and compute an anomaly score for each of them:

\[
s_a(\Pi(I)) = \frac{1}{n} \sum_{i=1}^{n} s_{\text{OOD}}((\Pi(I), \Pi_i), \phi \circ f_1) \tag{7}
\]

where \( s_{\text{OOD}} \) is an OOD score function which is presented in more detail in Section IV. In fact, we try different OOD functions and find out the best one. While \( n_{\text{sp}} \) permutations are randomly used during training, it is important to note that the \( n_{\text{sp}} \) permutations are fixed for all tests in the final model.

With this new piece-wise puzzle task, lower anomaly detection errors can be reached while keeping the same inference complexity as the partial puzzle task (see results in Fig. 11).

B. Tint Rotation Task

High-scale object colorimetry is a simple but powerful clue to discriminate anomalies, especially in spoof detection. To explore this rich information that is not considered yet in the literature, we present a novel tint rotation recognition task which focuses on the normal class colorimetry. Given an RGB image \( I \) and a transformation \( \gamma \) where \( \gamma(I, \theta) \) adds an offset \( \theta \) to the hue channel (in HSV space) of \( I \); we try to predict the distribution of \( \Theta \) from \( \gamma(I, \Theta) \). For practical reasons, we limit the possible tint rotation angles to \( c \) distributed angles and our task becomes to distinguish angles which are multiples of \( \frac{2\pi}{c} \).

Tackling the colorimetry task with a rotation detection task allows us to discriminatively learn high-scale and general colorimetry clues while keeping a low computational cost. In addition, we note that contrary to the geometrical rotation recognition task where a number of angles different from four would leave visual artifacts, our task does not have any limitation on \( c \).

Nevertheless it is impossible to detect any tint rotation inside areas without any original color information. To prevent high anomaly scores on desaturated images, we need to give a lower weight on those regions. To this end, instead of working on the angle distribution we use the expected \( L_1 \) error in RGB space between the original image and the predicted one. Since we are computing a pixel wise RGB error, only large areas of...
mix object pieces and background pieces. Solving tasks on task will allow our network to focus on specific image patches. Specialized dense layer for the pretext task. Each intra-piece new tasks during inference without increasing the number of potential ignored by the puzzle task in case of salient geometrical features.

C. Intra-Piece Tasks

On top of the piece-wise puzzle task, we further propose to add pretext sub-tasks inside each puzzle piece. Given an intra-piece task $T_{\text{piece}}$ and an image composed of $n$ pieces images $R_1, \ldots, R_n$, we first sample a random augmented piece using the pretext data generation function on each piece $(I_i^{\text{aug}}, y_i) \sim DG_{T_{\text{piece}}}(\{R_i\})$. Then our network tries to solve simultaneously the puzzle task and the intra-piece tasks by minimizing the loss

$$L(I) = \frac{1}{n} \sum \left( L_{\text{CE}}(\phi \circ f_i(I); \Pi_i) + L_{\text{piece}}(R_i) \right)$$

where the first term is from the piece-wise puzzle loss defined in Equation 6 and $L_{\text{piece}}$ is the loss of the intra-piece task. In our case, we choose the pretext rotation task for the intra-piece task thus $L_{\text{piece}} = L_{\text{int}}$. We argue that the piece-wise tint rotation task is more suitable than a piece-wise geometrical rotation task since it mixes different modalities rather than only combining geometrical cues. Besides, we have already studied the combination of jigsaw puzzle task with the geometric rotation in our previous work [9]. A summary of the intra-piece task model is given in Fig. 6.

By adding these intra-piece tasks, we essentially consider $n$ new tasks during inference without increasing the number of forward pass in our encoder. The only cost is the additional specialized dense layer for the pretext task. Each intra-piece task will allow our network to focus on specific image patches.

One issue with this method is that we can potentially mix object pieces and background pieces. Solving tasks on colorful pixels will impact the anomaly score. The tint rotation task training loss is:

$$L_{\text{int}}(y(I, \theta)) = E_{\theta} E_{y(I, \theta)} \left[ \frac{\|I - y(I, \theta - \Theta)\|_1}{W \times H \times 255} \right]$$

where $W \times H$ is the dimension of the image. For the anomaly score, we use the same error as the loss function which becomes in its developed form:

$$s_a(y(I, \theta)) = \sum_{i=1}^{c} \max(\phi \circ f(y(I, \theta)))_i \left( \frac{\|I - y(I, \theta - \Theta)\|_1}{W \times H \times 255} \right)$$

where $\max(\cdot)$ is the softmax function.

By introducing this task we force our encoder to fully represent the normal class colorimetry, which could be potentially ignored by the puzzle task in case of salient geometrical features.

background pieces would enable the model to generalize on image distribution far from the normal class object. As a result, we introduce a weight map for each piece learned during training where higher weights are given for pieces covering the object. We could see this map as a rough segmentation of the normal object in the image. These are computed in a similar fashion as visual attention mechanism, which have previously successfully been used for learning weight maps for each pixels [46].

First, we compute from the encoder representation $z$ a weight map $(w_{ij})_{0 \leq i+j \leq n}$, which we normalize into attention weights using the $L_1$ normalized sigmoid $P_{ij} = \sigma(w_{ij}) \frac{\|w_{ij}\|_1}{\sum_{i+j}}$. This normalization function produces smoother maps than the classical softmax activation, preventing very sparse maps where only one piece has a non-null activation. To further prevent these cases, we include an additional term to the loss encouraging spread matrices:

$$L_{\text{density}}(P) = \sum_{i,j} \left( i_j \right)^2 - \mu \|P_{ij}\|_2$$

where $\mu = \sum_{i,j} P_{ij} \left( i_j \right)^2$.

Our final loss of intra-piece tasks taking into account the attention map (upper branch in Fig. 3) is:

$$L_{U-branch} = L_{\text{pzl}}(I) + L_{\text{density}}(P) + \sum_{i,j} P_{ij} \cdot L_{\text{piece}}(R_{ij})$$

and the corresponding anomaly score is

$$s_a(I) = M(\|s_{OOD}(z, y) \circ f(I)\|_1, y) \in DG_T(\{I\}))$$

where $M$ is the fusion function which is detailed in Section IV.
We can see in Table IX that the attention mechanism increases anomaly detection performances.

**D. Partial Colorization Task**

We present in this section a novel generative pretext task for anomaly detection which is highly texture oriented. In the colorization task commonly used in the literature [31], [47], [48], the main objective is to predict the \((A, B)\) color channels from the luminance channel \(L\) of an image in LAB space.

One big challenge with this task is to colorize the background since it can vary a lot inside the training normal set. The re-colorization will be naturally poorer for unseen background during inference of new observations. Therefore the object itself should have more impact on the AD algorithms than the background, making the anomaly detector more object-oriented than scene-oriented. In addition, several issues arise when considering the typical framework of colorization through regression [47] where \(E[\{A_{ij}, B_{ij}\}]L\) is directly estimated for each pixel \((i, j)\). First, the colorimetry of the normal class can potentially be multi-modal. In other words, the normal class objects can have several plausible set of colors called modes. For example, horses could have more than one fur color yet still being part of the same class. In this case a regression network will end up predicting the mean of all modes ignoring the multi-modality. Second, even if one of the object mode is correctly predicted, any error function will yield high values if the mode of the current observation is different.

To tackle these limitations, we establish a novel method to learn colorization well-suited to anomaly detection. First, we augment the available inputs with the color values of the image inside a border of size \(\alpha\). We thus estimate a map \(y_{\text{part}}\) into a low-range discrete variable and perform multi-class classification.

**1) Color Bin Classification:** By quantizing each color value into \(K\) bins and assuming the two colors planes to be independent, we can define the resulting categorical variables by \(2K\) probabilities: \(P(A_{ij} = 1), \ldots, P(A_{ij} = K), P(B_{ij} = 1), \ldots, P(B_{ij} = K)\). We thus estimate a map \(y\) of dimension \(H \times W \times 2K\), where

\[
y_{i,j,2k} = P(A_{ij} = k|I_{\text{part}})
y_{i,j,2k+1} = P(B_{ij} = k|I_{\text{part}})
\]  

Inspired by the label smoothing idea [49], a gaussian smoothing is applied to the output distributions in order to propagate our model confidence to neighbor color bins. Indeed we do not want to entirely penalize close color bins. As such the final estimated density \(P(A_{ij} = k|I_{\text{part}})\) for a network \(\phi\) is

\[
\hat{P}(A_{ij} = k|I_{\text{part}}) = (\text{max}(\phi(I_{\text{part}})_{ij}) \ast G_\alpha)_k
\]

where \(G_\alpha\) is the gaussian kernel of standard deviation \(\sigma\).

**2) Gaussian Mixture Model MLE:** Our second approach is to parameterize the densities with Gaussian Mixture Models. Accordingly, we have for each pixel a sum of \(K\) gaussian densities:

\[
p(A_{ij}, B_{ij}|I_{\text{part}}) = \sum_{k=1}^{K} \pi_{ij}^{(k)} \mathcal{N}(A_{ij}, B_{ij}; \mu_{ij}^{(k)}, \Sigma_{ij}^{(k)})
\]

where \(\pi_{ij}^{(k)} \in \mathbb{R}\) is the prior probability of the \(k\)th cluster, \(\mu_{ij}^{(k)} \in \mathbb{R}^2\) is the mean color of the \(k\)th cluster and \(\Sigma_{ij}^{(k)} \in \mathbb{R}^{2 \times 2}\) is the covariance color matrix of the \(k\)th cluster.

Rather than predicting the full \(2 \times 2\) matrix \(\Sigma_{ij}^{(k)}\), we only predict the three free parameters \(\sigma\). We can then reconstruct the positive definite covariance matrix using Cholesky decomposition [50]:

\[
\Sigma_{ij}^{(k)} = \begin{bmatrix} 1 & 0 \\ 0 & l \end{bmatrix} \text{Diag} \left( e^d \right) \begin{bmatrix} 1 & 0 \\ 0 & l \end{bmatrix}^T
\]

where \(d \in \mathbb{R}^2\) and \(l \in \mathbb{R}\). This decomposition ensures strictly positive eigen values from the exponential and a semi-positive matrix from the Cholesky decomposition. All the possible covariance matrices are thus parameterized by \((d, l)\). It also introduces better numerical stability for determinant computation with the simple formula

\[
\log |\Sigma| = \log |\text{Diag}(e^d)| = \sum d_i.
\]

To train this model, we could use as the loss function the log-likelihood which considers all pixels independent:

\[
\mathcal{L}(\mu, \Sigma|A, B) = \sum_{ij} \log \left( \sum_{k=1}^{K} \pi_{ij}^{(k)} \mathcal{N}(A_{ij}, B_{ij}; \mu_{ij}^{(k)}, \Sigma_{ij}^{(k)}) \right)
\]

However this function turns out to be very hard to directly optimize for each pixel and does not lead to any meaningful colorization. We use instead the classical Expectation Maximization algorithm. As for details, we carry out the three following steps:

1) **Compute Mahalanobis distances:**

\[
\Delta_{ij}^{(k)} = \left( l_{ij} - \mu_{ij}^{(k)} \right)^T \Sigma_{ij}^{(k)-1} \left( l_{ij} - \mu_{ij}^{(k)} \right)
\]

2) **Compute posterior cluster probabilities:**

\[
\gamma_{ij}(k) = \frac{\pi_{ij}^{(k)} \exp \left( -\frac{1}{2} \left( \sum d_i + \Delta_{ij}^{(k)} \right) \right)}{\sum_{k=1}^{K} \pi_{ij}^{(k)} \exp \left( -\frac{1}{2} \left( \sum d_i + \Delta_{ij}^{(k)} \right) \right)}
\]

3) **Fix the \(\gamma_{ij}(k)\) and minimize loss (lower branch):**

\[
\mathcal{L}_L(\pi, \mu, \Sigma|I) = \sum_{ij} \sum_{k=1}^{K} \gamma_{ij}(k) \left( \Delta_{ij}^{(k)} + \sum d_i^{(k)} - \log \pi_{ij}^{(k)} \right)
\]
### Algorithm 1 Our Model Training

1. **Input**: batch size \( B \)
2. **Initialization**: upper-branch encoder \( \phi \), task-specific networks \( f_{\text{pal}}, f_{\text{int}}, f_{\text{att}} \), attention network \( f_{\text{att}} \), U-shape enc-dec \( \psi \)
3. **while** not reach the maximum epoch do
   4. Sample image minibatch \( x \)
   5. Transform batch to \( n_{\text{step}} \) shuffled images \( x'_{1}, \ldots, x'_{n_{\text{step}}} \) with piecewise tint rotation
   6. **for** \( k = 1 \cdots n_{\text{step}} \) do
      7. Apply encoder \( z_k \leftarrow \phi(x'_k) \)
      8. Compute puzzle loss \( L_{\text{pal}} \) from Eq.6
      9. Compute tint loss \( L_{\text{int}} \) from Eq.8 with attention
   **end for**
   10. Decolorize batch to \( x_{\text{decolor}} \)
   11. Perform EM algorithm from Eq.19,20,21
   12. Gradient descent on \( L_{\text{U-branch}} \) to update \( \phi, f_{\text{pal}}, f_{\text{int}}, f_{\text{att}} \)
   13. Gradient descent on \( L_{\text{L-branch}} \) to update \( \psi \)
   **end while**
16. **Output**: networks \( \phi, \psi, f_{\text{pal}}, f_{\text{int}}, f_{\text{att}} \)

Once the training is finished, we compute the anomaly score as the likelihood of the color channels under the predicted \( \pi_{ij}^{(k)}, \mu_{ij}^{(k)} \) and \( \Sigma_{ij}^{(k)} \):

\[
s_{a}(A_{ij}, B_{ij}|I_{part}) = \sum_{k=1}^{K_1} \pi_{ij}^{(k)} \mathcal{N}(A_{ij}, B_{ij}; \mu_{ij}^{(k)}, \Sigma_{ij}^{(k)})
\]

(22)

In order to choose the number of gaussians \( K \), we apply beforehand a K-means color clusterization [51] on the cropped down-sampled images of the normal class. Then by using the elbow method, we can find the optimal \( K \) inside \([1, 10]\).

**Advantages**. The GMM approach has three advantages over the bin classification: (i) its density support is not bounded, and is continuous thus not needing any gaussian smoothing, (ii) it can fully model the dependence between the color channels with the full covariance matrix, and (iii) it can reach the same quality of colorization with fewer parameters. The quality of colorization is here measured using the mean pixel color likelihood.

### IV. OOD METHODS AND FUSION

We try two different out-of-distribution methods for each pretext task: the softmax truth and the Mahalanobis distance. In the case of a self-supervised classification task, the most commonly used OOD function is the likelihood of the label given that the image is normal, which we call the “softmax truth”:

\[
s_{\text{OOD}}((z, y); \phi \circ f) = p(y|z, z \in X_{\text{train}})
\]

(23)

However, this softmax truth criterion takes into account only one component of the softmax vector. For easy tasks, we usually have a high probability on the correct class, however for harder, multi-issue task, we can have several typical highly activated classes for the normal class. As such, another idea is to look at the likelihood of the raw score vector given its label and given that the image is normal:

\[
s_{\text{OOD}}((z, y); \phi \circ f) = p(\phi \circ f(z))|y, z \in X_{\text{train}}
\]

(24)

To approximate this conditional probability, the training dataset is first partitioned on samples sharing the same label value \( l \), i.e. \( \{(z, y)|l(y) \in \mathcal{X}_{\text{train}} \} \). The distribution of the normal class raw score vectors given \( y \) can then be separately estimated on each partition after convergence of the network weights.

For a given classification problem with \( C \) classes and a training set \( \mathcal{X}_{\text{norm}} \), we estimate the mean scores \( \mu_{c} \) and covariance matrices \( \Sigma_{c} \) for each class \( c \):

\[
\mu_{c} = \frac{1}{|Z_{c}|} \sum_{z \in Z_{c}} \phi \circ f(z)
\]

\[
\Sigma_{c} = \frac{1}{|Z_{c}|} \sum_{z \in Z_{c}} (\phi \circ f(z) - \mu_{c})^{2}
\]

(25)

where \( Z_{c} = \{z|l(y) \in \mathcal{D}G_{T}(\mathcal{X}_{\text{norm}}) \} \) and \( y = c \). The OOD score is approximated by the Mahalanobis distance [52] with the mode corresponding to the truth label:

\[
s_{\text{OOD}}((z, y); \phi \circ f) \approx (\phi \circ f(z) - \mu_{y})^{T} \Sigma_{y}^{-1}(\phi \circ f(z) - \mu_{y})
\]

(26)

We also explore different fusion functions to combine all the OOD scores into a single anomaly score. We first use the mean, but observe heavy biases from outlier OOD scores (very easy sub-task or harder sub-task). We then try different order statistics including the median and the 25th percentile and compare the results in Table VII.

### V. FULL METHOD OVERVIEW

This section summarizes our full method (Fig. 3). Our model is made of two independent branches. The first discriminative branch (upper branch in Fig. 3) solves the piece-wise puzzle task with intra-piece tint rotation detection task. The second generative branch (lower branch in Fig. 3) performs the partial re-colorization task. We share the same encoder network for all of the discriminative tasks, including the attention mechanism. The re-colorization task is modeled...
Algorithm 2: Our Model Inference

1. Input: image \( x \)
2. Transform input to \( n_{sp} \) shuffled images \( x'_1, \ldots, x'_n_{sp} \) with piecewise tint rotation
3. for \( k = 1 \cdots n_{sp} \) do
4. Apply encoder \( z_k \leftarrow \phi(x'_k) \)
5. Compute \( s_{\text{pumsk}} \) from Eq. 7
6. Compute \( s_{\text{intk}} \) from Eq. 9
7. end for
8. Decolorize input to \( x_{\text{decolor}} \)
9. Compute \( \mu, \Sigma \) and \( \pi \) using the U-shape enc.-dec. on \( x_{\text{decolor}} \)
10. for \( i = 1 \cdots H, j = 1 \cdots W \) do
11. Compute \( s_{\text{color},i,j} \) from Eq. 22
12. end for
13. \( s_a \leftarrow \text{median}(\text{median}(s_{\text{pumsk}}), \text{median}(s_{\text{intk}}), \text{median}(s_{\text{color},i,j})) \)
14. Output: Anomaly score \( s_a \)

with GMM, and we include the attention mechanism for the intra-piece task. To detect whether or not an observation \( x \) is an anomaly, we produce the OOD scores of the re-colorization and the \( n_{sp} \) sampled permutations along with tint rotation tasks. The chosen OOD function for every task is the softmax truth. All of these scores are then combined into a single anomaly score using the median.

Our full training and inference algorithms are respectively given in Alg. 1 and Alg. 2.

Presented in Fig. 9 are examples of anomalies detected by our three different tasks using different visual cues. As can be seen, our detectors are of complementary strength.

VI. RESULTS

A. Evaluation Protocol

Our evaluation protocol is made of three types of anomaly detection challenges: object anomalies, fine-grained style anomalies, and face presentation attacks. First, to detect object anomalies we use general coarse object recognition datasets. The one-vs-all protocol is used, where we consider one class of a multi-classification dataset as the normal class. All the other classes are then considered as anomalous, and we can obtain a set of runs for each possible normal class. Thus, for a given run the training dataset is the normal class training data and the test dataset contains the original test data of the normal class and the anomalous classes. The final reported result is the mean of all runs.

However, these datasets have become far from real anomaly detection applications and might not be enough to fully evaluate AD methods. Thus we include a second evaluation group where we try to detect style anomalies using fine-grained classification datasets. Fine-grained datasets have been introduced to tackle the recognition of classes, usually part of a same category, with slight differences. We use here the one-vs-all protocol as well.

Finally, we consider a real anomaly detection problem which incorporates object anomalies, style anomalies and local anomalies. In particular we choose a dataset from face presentation attack detection (FPAD), where the goal is to discriminate real faces from fake representations of someone’s face. Due to the constantly evolving frauds and high variability, anomaly detection seems a very appealing solution to this problem.

We use the following datasets:

(i) For **object anomalies**:
- **F-MNIST** [53]: has been introduced as a harder version of MNIST with 10 different classes of fashion items. All images are grayscale meaning no color information can be used to discriminate anomalies.
- **CIFAR-10** [54]: object recognition dataset composed of 10 wide classes with 6000 images per class.
- **CIFAR-100** [54]: extended version of CIFAR-10 with 100 classes each containing 600 images.

(ii) For **style anomalies**:
- **Caltech-UCSD Birds 200** [55]: fine-grained classification dataset of 200 birds species with approximately 30 images per class.
- **FounderType-200** [56]: font recognition dataset containing 200 fonts with 6700 images per class. It has been introduced for novelty detection and even though these images lie on a low dimensional manifold compared to natural images, they still provide insight into how well the model can capture small shape hints.

(iii) For **face presentation attack detection**, we use the WMCA dataset [57] which contains more than 1900 short videos of real faces and presentation attacks. It contains several modalities such as infra-red or depth, but here we only use RGB. There are 72 real identities along with several types of attacks: paper print, screen replay, masks and partial attacks where only a localized area of the face is fake. The masks are composed of paper masks, rigid mask and flexible masks. An example of each type of attack is given in Fig. 10.

In all evaluations, the metric used is the area under the ROC curve (AUROC) or the error 1-AUROC, averaged over all possible normal classes in the case of one-vs-all datasets.

### Table II

| Dataset       | Anomaly type | Object | Style | Local |
|---------------|--------------|--------|-------|-------|
| F-MNIST       | ✓            | -      | -     |       |
| CIFAR-10      | ✓            | -      | -     |       |
| CIFAR-100     | ✓            | -      | -     |       |
| Fine-grained  |              |        | ✓     | -     |
| Caltech-Birds |              | ✓      |       | -     |
| FounderType   |              | -      | ✓     | -     |
| FPAD          |              | ✓      | ✓     | ✓     |

**Fig. 9.** Examples of anomalies when considering one-vs-all on CIFAR-10. We indicate if each task detects it as an anomaly (✓) or as normal (−).
We additionally include for anti-spoofing datasets metrics more adapted to biometric presentation attack detection:

- The equal error rate (EER [58]), which is the location in the ROC curve where the false reject rate (or Bona-fide Presentation Classification Error Rate BPCER) is equal to the false acceptance rate (or Attack Presentation Classification Error Rate APCER).
- The Attack Presentation Classification Error Rate for the Bona-fide Presentation Classification Error Rate fixed at 5% (APCER@5%BPCER [58]).

B. Implementation Details

For the piece-wise puzzle task, we use a margin of half the size of the pieces and find best results with \( n_{sp} = 18 \). Generally we use \( n_w = n_h = 3 \) pieces for most datasets, except face anti-spoofing where \( n_w = 3 \) and \( n_h = 4 \). We observe better results with more vertical pieces on faces, since they are always upright and need finer vertical analysis. For the tint rotation recognition we use \( c = 4 \) and for the re-colorization task, we use a contextual border \( \alpha \) of two pixels.

Regarding network architecture, we use a 16-4 WideResNet [61] (\( \approx 10M \) parameters with a depth of 16) for the feature extractor network \( \phi \), along with three dense layers respectively of size \( n^2 \) for the piece-wise puzzle task, size \( n \cdot c \) for the tint rotation task and size \( n \) for the attention. Each of these dense layers have a dropout rate of 0.3 [68]. As for the re-colorization task, we use a UNet network [69]. It was originally introduced for image segmentation, using a down-sample / up-sample strategy reintroducing the intermediate maps at each step of the down-sample branch into the up-sample branch. It is in fact generally well suited for any prediction task where the output is aligned with the input pixels (in our case a vector of GMM parameters for each pixel). Training is performed under SGD optimizer with Nesterov momentum [70], using a batch size of 32 and a cosine annealing learning rate scheduler [71].

C. Comparison to the State-of-the-Art

A comparison of our method with other state-of-the-art (SOTA) anomaly detection models is performed on all three protocols. We choose to include three families of SOTA methods: one-class learning methods which only learn using the normal class, semi-supervised methods where a small set of anomalies is used during training and supervised learning. The considered one-class methods can be categorized into (1) reconstruction error-based methods with ADGAN [59], GANomaly [22] and PIAD [23], (2) hybrid methods with OC-SVM [13], IF [16], OC-CNN [19], (3) pretext tasks-based methods with ARNet [60], GeoTrans [8], MHRot [37] and PuzzleGeom [9] and (4) two-stage anomaly detection using contrastive learning with SSD [40] and DROC-contrastive [38]. GeoTrans uses various geometrical transformations as SSL pretext task, MHRot adds on top 90° rotations and our previous model PuzzleGeom [9] includes a basic jigsaw puzzle task. Regarding semi-supervised methods, we evaluate DeepSAD [26] trained on the same normal samples but with three different ratio of the anomaly subclasses: 10%, 25% and 75%. For the fully supervised baseline we simply use the same backbone as our one-class method (the 16-4 WideResNet) extended with a dense layer representing the two normal and anomaly classes. It is important to note that its training is performed with classical binary cross-entropy loss on the normal/anomaly label, without any class balancing mechanism.

The experiment results are displayed in Table III and a detailed evaluation on the CIFAR-10 dataset is included in Table IV. We note that for the sake of fair comparison in the same conditions, we re-evaluate almost all methods ourselves using existing implementations.

Our method maintains among the best accuracies on coarse object and fine-grained anomaly detection. It improves upon PuzzleGeom, and closes the gap toward semi-supervised performances with a small AUC difference of 0.5% on CIFAR-100. Compared to previous pretext tasks such as rotation detection, our proposed tasks can better focus on local parts of the image. The re-colorization task will target more fine-grained local textures while the puzzle task and intra-piece tint detection will work on higher-scale geometrical and colorimetric features of the image. We also show that our method greatly improves anti-spoofing detection performance on WMCA. It even outperforms the supervised model and semi-supervised anomaly detection methods which have access up to 75% of the anomalous data.

In general we can notice that hybrid methods, although efficient for smaller problems, do not extend well to high-dimensional data. The evaluated reconstruction-based methods also tend to fall behind pretext-task oriented models. On the other hand, two-stage contrastive methods like DROC-contrastive produce very competitive performance. This model combines different techniques including contrastive representation learning, distribution augmentation and OC-SVM. It performs slightly better than ours on the F-MNIST dataset and reaches the same AUC on CIFAR-10 but on the more challenging one, CIFAR-100, we obtain a gain of nearly 2%. Moreover, we note that distribution augmentation and OC-SVM could also be used on the concatenation of our learned representations to reach better accuracy.
TABLE III
Comparison with the state-of-the-art AUROC over several datasets, underline indicates best result, bold indicates best one-class learning result. For the sake of fair comparison, we re-evaluated by ourselves all methods, except the one-class methods in the first block. (Results are from [38], [59], [60]). DROC-Contrastive [38] combines different techniques:
Contrastive learning, distribution augmentation and OC-SVM

| Model          | CIFAR-10 | CIFAR-100 | F-MNIST | CUB-200 | FounderType | WMCA |
|----------------|----------|-----------|---------|---------|-------------|------|
| Supervised     |          |           |         |         |             |      |
| I6-I WideResNet [61] | 99.3     | 96.3      | 99.2    | -       | -           | 82.4 |
| Deep-SAD (75%) [26]  | 92.5     | 88.7      | 98.1    | 73.6    | 99.8        | 83.2 |
| Deep-SAD (25%)     | 90.8     | 87.9      | 95.4    | 70.9    | 99.4        | 79.8 |
| Deep-SAD (10%)     | 86.0     | 89.1      | 88.2    | 66.1    | 98.0        | 72.6 |
| Semi-Supervised   |          |           |         |         |             |      |
| ADGAN [59]        | 62.4     | 54.7      | 88.4    | -       | -           | -    |
| GANomaly [22]     | 69.5     | 56.5      | 80.9    | -       | -           | -    |
| ARNet [60]        | 86.6     | 78.8      | 93.9    | -       | -           | -    |
| DROC-contrastive [38] | 92.5     | 86.5      | 94.8    | -       | -           | -    |
| One-class         |          |           |         |         |             |      |
| OCSVM [13]        | 58.5     | -         | 74.2    | 76.3    | -           | -    |
| IF [16]           | 73.4     | -         | 84.0    | 74.2    | -           | -    |
| OC-CNN [19]       | 66.5     | -         | 75.4    | -       | -           | -    |
| PIAD [23]         | 79.9     | 78.8      | 94.3    | 63.5    | 90.8        | 76.4 |
| GeoTrans [8]      | 85.4     | 84.7      | 92.6    | 66.6    | 92.3        | 79.8 |
| MHRot [37]        | 89.5     | 83.6      | 92.5    | 77.6    | 96.7        | 81.3 |
| PuzzleGeom [9]    | 88.2     | 85.8      | 92.8    | 83.2    | 96.9        | 85.6 |
| **Ours**          | 92.5     | 88.2      | 93.7    | 83.2    | 97.4        | 91.4 |

TABLE IV
Detailed comparison with one-class state-of-the-art AUROC on the CIFAR-10 dataset

| Model   | Airplane | Automobile | Bird | Cat | Deer | Dog | Frog | Horse | Ship | Truck | Avg  |
|---------|----------|------------|------|-----|------|-----|------|-------|------|-------|------|
| VAE [62] | 70.0     | 38.6       | 67.9 | 53.5| 74.8 | 52.3| 68.7 | 49.3  | 69.6 | 38.6  | 58.3 |
| OCSVM [13] | 63.0     | 44.0       | 64.9 | 48.7| 73.5 | 50.0| 72.5 | 53.3  | 64.9 | 50.8  | 58.5 |
| AnoGAN [21] | 67.1     | 54.7       | 52.9 | 54.5| 65.1 | 60.3| 58.5 | 62.5  | 75.8 | 66.5  | 61.8 |
| PixelCNN [63] | 53.1     | 99.5       | 47.6 | 51.7| 73.9 | 54.2| 59.2 | 78.9  | 34.0 | 66.2  | 61.8 |
| Deep-SVDD [64] | 61.7     | 65.9       | 50.8 | 59.1| 60.9 | 65.7| 67.7 | 67.3  | 75.9 | 73.1  | 64.8 |
| OCGAN [65] | 75.7     | 53.1       | 64.0 | 62.0| 72.3 | 62.0| 72.3 | 57.5  | 82.0 | 55.4  | 65.6 |
| Puzzle-AE [66] | 78.9     | 78.0       | 69.9 | 54.8| 75.4 | 66.0| 74.7 | 73.3  | 83.3 | 69.9  | 72.4 |
| DROCC [20] | 81.7     | 76.7       | 66.7 | 67.1| 73.6 | 74.4| 74.4 | 71.4  | 80.0 | 76.2  | 74.2 |
| AnoNAGN [67] | 96.2     | 63.8       | 72.5 | 64.3| 87.3 | 63.8| 88.3 | 58.4  | 93.5 | 64.5  | 75.01 |
| GeoTrans [8] | 74.7     | 95.7       | 78.1 | 72.4| 87.8 | 87.8| 83.4 | 95.5  | 93.3 | 91.3  | 86.0 |
| PuzzleGeom [9] | 75.1     | 96.3       | 84.8 | 74.2| 91.1 | 89.9| 88.7 | 95.5  | 94.7 | 91.9  | 88.2 |
| SSD [40]   | 82.7     | 98.5       | 84.2 | 84.5| 84.8 | 89.9| 91.7 | 95.2  | 92.9 | 94.4  | 90.0 |
| **Ours**  | 85.9     | 97.9       | 88.7 | 81.2| 95.4 | 94.2| 92.1 | 96.9  | 96.5 | 95.4  | 92.5 |

TABLE V
AUROC, EER and APCER at 5% BPCER on WMCA dataset, best result is in bold

| Models     | AUROC | EER  | APCER (5%BPCER) |
|------------|-------|------|-----------------|
| MHRot [37] | 81.3  | 23.9 | 72.6%           |
| PuzzleGeom [9] | 85.6  | 19.7 | 33.8%           |
| **Ours**   | 91.4  | 16.1 | **27.3%**       |

Overall, our model keeps a good balance between coarse object anomaly detection and finer style anomaly detection, and even outperforms semi-supervised anomaly detection methods on CUB-200 and WMCA. It achieves a relative error improvement of 36% on CIFAR-10 and 40% on WMCA compared to PuzzleGeom.

Lastly, we compare in Table V our method with the second best self-supervised methods MHRot and PuzzleGeom on WMCA. Using our method the APCER@5%BPCER drops from 33.8% to 27.3%. This also shows promising usage of anomaly detection methods in fraud detection.

VII. PARAMETER STUDY

In this section, we evaluate the parametrization of pretext tasks in Sections VII-A, VII-B, VII-C, the choice of OOD function in Section VII-D and perform an ablation study in Section VII-E.

A. Puzzle Task Complexity

We start by comparing in Fig. 11 the two approaches on the CIFAR-10 dataset for the jigsaw puzzle task introduced in Section III-A. The piece-wise puzzle task greatly improves performances for all CIFAR-10 classes even though the same permutations are tested during inference. Moreover, we confirm that the partial puzzle task is more sensitive to the choice of \( n_{sp} \), since its representation quality also depends on this factor. We choose to fix \( n_{sp} = 18 \) out of 9! possible
permutations as a good compromise between complexity of inference and accuracy.

The influence of the number of puzzle pieces \( n_w \) and \( n_h \) for \( n_{sp} \in \{9, 18\} \) is reported in Fig. 12 on CIFAR-10. We can see that for both \( n_{sp} = 9 \) and \( n_{sp} = 18 \), the best value for general one-vs-all problem is \( n_w = n_h = 3 \).

B. Tint Rotation Task Complexity

We measure the AUC of the isolated tint rotation task for different number of tint rotations \( c \) on the CIFAR-10 dataset in Fig. 13. The best value of \( c \) across several normal classes is 4.

C. Colorization Task Parametrization

The two colorization parametrizations using Gaussian Mixture Model and bin classification are compared on the normal class full colorization task. Our evaluation metric is directly the likelihood of the colorization, which is respectively

\[
L(A, B) = \prod_{i,j} \text{smax}(\phi(I)_{ij}, \frac{A_{ij}}{\sum_k B_{ij} n_{ij}}) \cdot \text{smax}(\phi(I)_{ij}, \frac{n_{ij}}{\sum_k B_{ij} n_{ij}})
\]

for classification and GMM

\[
L(A, B) = \prod_{i,j} \sum_k n_{ij}^{(k)} \mathcal{N}(A_{ij}, B_{ij}; \mu_{ij}^{(k)}, \Sigma_{ij}^{(k)})
\]

Overall, we can reach higher likelihoods with GMM than bin classification. Moreover, a better separation of the different modes can be achieved using GMM, where bin classification usually mixes the different modes and produces dull colors (see Fig. 14).

D. Choice of OOD and Fusion Functions

To evaluate the effect of Mahalanobis distance as an anomaly score, we compare it with the softmax truth and its improved form, the ODIN method [72] which adds temperature scaling during training, and the input pre-processing \( \tilde{x} = x - \varepsilon \text{sign}(-\nabla_x \log \text{softmax}(x; T)) \).

The results are presented in Fig. 15 for different number of puzzle pieces \( n \) and \( n_{sp} = 18 \) permutations tested. The AUC increases with the number of pieces when using the Mahalanobis distance, whereas it decreases with the softmax truth. In addition, the AUC of the most difficult class is always higher when using the Mahalanobis distance. This shows that despite a lower average anomaly detection performance, it has less variance in its predictions and provides more robust OOD scores to different normal classes. Even though the ODIN method provides sensible improvement for more than \( 3 \times 3 \) pieces, it greatly increases computational complexity during training and inference. In our tests, we observe an inference time increase of more than three times with the ODIN method. We provide in Table VI further comparisons between the softmax truth and the Mahalanobis distance on the puzzle task with \( n_{sp} = 9 \).

Finally, we evaluate the choice of different fusion functions on the WMCA dataset in Table VII. The evaluated fusion
functions are simple order statistics commonly found among ensemble learning decision fusion strategies. We observe overall better performances regarding AUC and APCER with the median fusion function.

**E. Ablation Study**

We evaluate the impact of each pretext task on the final anomaly detection AUROC. In Table VIII, we compare on CIFAR-10 the basic partial puzzle model with the addition of the piece-wise puzzle task, colorization task, intra-piece tint rotation detection task with and without the attention map. While the piece-wise puzzle and colorization give our model great discrimination power with an AUC of 89.12, the intra-piece task with attention further refines our model.

We also investigate on more datasets how the addition of attention in the intra-piece task improves anomaly detection in Table IX. By including attention weights for each piece, we can further improve the mean AUC on all datasets, although marginally increasing the prediction variances on different normal classes. We can also notice that the usage of attention has varying contribution depending on the dataset. The main role of the attention for the intra-piece task is to prevent our task-specific model to generalize too much on background pieces. Thus, attention will benefit the most when the normal class background is very diverse or the normal object is very small in the image.

**VIII. Conclusion and Future Work**

We explore in this paper more efficient pretext tasks and show that a combination of a colorization and a puzzle task with intra-piece tint rotation subtasks provides the best anomaly detection performances. We also show the importance of different out-of-distribution functions along with their fusion functions. Finally, we provide a more comprehensive evaluation protocol than previously used datasets in the anomaly detection literature. It presents more challenging datasets and covers object, style and local anomalies. Our method outperforms state-of-the-art, including a semi-supervised method, on most of the fine-grained datasets.

For future work we could explore other generative pretext tasks such as image reconstruction. As in the colorization task, only a part of the image mostly covering the normal object would be destroyed. Furthermore, generative tasks such as our current colorization could be used to locate anomalies using the pixel-wise error. Finally we could reframe our method into a two-stage anomaly detection. In a first step, representations would be learned solving our pretext re-colorization, jigsaw puzzle and intra-piece tint rotation detection tasks. Then we could separately train a OC-SVM on the concatenation of representations from the puzzle and colorization encoder. We could further evaluate our model with differently sized backbones and measure the impact on each of our three pretext tasks.
Loïc Jézéquel (Graduate Student Member, IEEE) received the Master of Engineering degree from Télécom SudParis in 2019. He is currently pursuing the Ph.D. degree with the ETIS Laboratory—CY Cergy Paris University, ENSEA, CNRS, and IDEMIA Identity and Security. His research focuses on anomaly detection and face anti-spoofing with deep learning.

Ngoc-Son Vu received the Ph.D. degree from UGA-Université Grenoble Alpes in 2010. He was a Postdoctoral Researcher at INSA de Lyon and Grenoble-INP. He is currently an Associate Professor with the ETIS Laboratory—CY Cergy Paris University, ENSEA, CNRS, France. His research interests include computer vision and deep learning. He was awarded two best paper awards at the IEEE CBMI 2016 and IEEE IJCB 2011 conferences.

Jean Beaudet. photograph and biography not available at the time of publication.

Aymeric Histace is currently a Professeur des Universités with ENSEA, French Graduated School of Engineering (Bac+5). He is also the Deputy Director of the School, and the Head of Research, Innovation and Partnerships. He does research in computer vision, signal and image processing in interaction with natural science, engineering, medicine, and information science at the ETIS Laboratory (UMR 8051, ENSEA, UCP, CNRS). He is the Head of the CELL Team which activities is mostly dedicated to smart, reliable, and reconfigurable embedded systems for various domains.