Multiresolution Knowledge Distillation for Anomaly Detection

Mohammadreza Salehi, Niousha Sadjadi*, Soroosh Baselizadeh*, Mohammad H. Rohban, Hamid R. Rabiee
Department of Computer Engineering, Sharif University of Technology
(smrsalehi, nsadjadi, baselizadeh)@ce.sharif.edu, (rohban, rabiee)@sharif.edu

Abstract

Unsupervised representation learning has proved to be a critical component of anomaly detection/localization in images. The challenges to learn such a representation are two-fold. Firstly, the sample size is not often large enough to learn a rich generalizable representation through conventional techniques. Secondly, while only normal samples are available at training, the learned features should be discriminative of normal and anomalous samples. Here, we propose to use the “distillation” of features at various layers of an expert network, which is pre-trained on ImageNet, into a simpler cloner network to tackle both issues. We detect and localize anomalies using the discrepancy between the expert and cloner networks’ intermediate activation values given an input sample. We show that considering multiple intermediate hints in distillation leads to better exploitation of the expert’s knowledge and a more distinctive discrepancy between the two networks, compared to utilizing only the last layer activation values. Notably, previous methods either fail in precise anomaly localization or need expensive region-based training. In contrast, with no need for any special or intensive training procedure, we incorporate interpretability algorithms in our novel framework to localize anomalous regions. Despite the striking difference between some test datasets and ImageNet, we achieve competitive or significantly superior results compared to SOTA on MNIST, F-MNIST, CIFAR-10, MVTecAD, Retinal-OCT, and two other medical datasets on both anomaly detection and localization.

1. Introduction

Anomaly detection (AD) aims for recognizing test-time inputs that look abnormal or novel to the model according to previously seen normal samples during training. AD has been a vital demanding task in computer vision with various applications, like in industrial image-based product quality control [27, 7], or health monitoring processes [26]. These tasks also require pixel-precise localization of anomalous regions, called defects. This is pivotal for comprehending the dynamics of monitored procedures, triggering the apt antedotes, and providing pertinent data for downstream models in industrial settings.

Traditionally, the AD problem has been approached in a one-class setting, where anomalies represent a broadly different class from normal samples. Recently, considering subtle anomalies has attracted attention. This new setting further necessitates precise anomaly localization. However, performing excellently in both settings on various datasets is highly appreciated but is not fully achieved.

Due to the unsupervised nature of the AD problem and the restricted data access, only having anomaly-free data in training, the majority of AD methods [36, 31, 40, 18, 34] model the normal data abstraction by extracting semantically meaningful latent features. These methods perform...
well solely on either of the two mentioned cases. This problem, called the \textit{generality} problem [39], highly declines trust in them on unseen future datasets. Moreover, anomaly localization is either impossible inadequate in most of them [36, 31, 33] and leads to intensive computations that hurt their real-time performance. Additionally, many earlier works [33, 31] suffer from unstable training, requiring unprincipled early stopping to achieve acceptable results.

Though not fully explored in the AD context, using pre-trained networks could potentially be an alternative track. This is especially helpful when the sample size is small and the normal class shows significant variations. Some earlier studies [4, 12, 28, 29] try to train their model based on pre-trained features of the normal data. These methods either miss anomaly localization [4, 12], or tackle the problem in a region-based fashion [28, 53], i.e., splitting images into smaller patches to determine the sub-regional abnormality. This is computationally expensive and often leads to inaccurate localization. To evade this issue, Bergmann \textit{et al.} [8] train an ensemble of student networks to mimic the \textit{last layer} of a teacher network on the anomaly-free data. However, performing a region-based approach in this work makes it heavily rely on the size of the cropped patches and hence susceptible to the changes in this size, and intensifies the training cost severely. Furthermore, imitating only the last layer misses fully exploiting the knowledge of the teacher network [32]. This makes them complicate their model and employ other complementary techniques, such as self-supervised learning in parallel.

Lately, Zhang \textit{et al.} [52] have demonstrated that activation values of intermediate layers of neural networks are a solid perceptual representation of their input images. By this premise, we propose a novel knowledge distillation method for AD that is designed to \textit{distill} the \textit{comprehensive} knowledge of an ImageNet pre-trained \textit{source} network, \textit{solely} on the normal training data, into a simpler \textit{cloner} network. This happens by forcing the cloner’s \textit{intermediate} embedding of normal training data at \textit{several critical layers} to conform to those of the source. Consequently, the cloner learns the normal data manifold thoroughly and yet earns no knowledge from the source about other possible input data. Hence, the cloner will behave differently from the source when fed with the anomalous data. Furthermore, a simpler cloner architecture enables avoiding distraction by non-distinguishing features and enhances the discrepancy in the behavior of the two networks on anomalies.

Moreover, we derive precise anomaly localization heat maps without using region-based expensive training and testing through exploiting the concept of gradient. We evaluate our method on a comprehensive set of datasets on various anomaly detection/localization tasks, where we exceed SOTA in both localization and detection. Our training is highly stable and needs no dataset-dependent fine-tuning. As we only train the cloner’s parameters, we require just one more forward pass of inputs through the source compared to a standard network training on the normal data. We also investigate our method through exhaustive ablation studies. Our main contributions are summarized as follows:

1. Enabling a more comprehensive transfer of the knowledge of the pre-trained expert network to the cloner one. Distilling the knowledge into a more compact network also helps to concentrate solely on the features that are distinguishing normal vs. anomalous.

2. Our method has a computationally inexpensive and stable training process compared to the earlier work.

3. Our method allows a real-time and precise anomaly localization based on computing gradients of the discrepancy loss concerning the input.

4. Conducting a considerable number of diverse experiments and outperforming previous SOTA models by a large margin on many datasets and yet staying competitive on the rest.

2. Related Work

Previous Methods: Autoencoder (AE)-based methods use the idea that abnormal inputs are not reconstructed as precisely as normal ones by learning normal latent features. Hence, anomalous samples will have higher reconstruction errors than normal ones. To better learn these normal latent features, LSA [1] trains an autoregressive model at its latent space, and OC-GAN [31] attempts to force abnormal inputs to be reconstructed as normal ones. These methods fail on industrial or complex datasets [38]. SSIM-AE [10] trains an AE with the SSIM loss [54] instead of MSE, causing it to perform just better on defect segmentation. Gradient-based VAE [15] introduces an energy criterion, which is minimized at test-time by an iterative procedure. Both of the latter methods do not perform well on one-class settings, such as CIFAR-10 [23].

GAN-based approaches, like AnoGan [41], f-AnoGan [40], and GANomaly [3], attempt to find a specific latent space where the generator’s reconstructions, obtained from samplings of this space, are analogous to normal samples. f-AnoGan and GANomaly add an extra encoder to the generator to reduce the inference time of AnoGan. Despite their acceptable performance in localization and detection of subtle anomalies, they fail in one-class settings.

Methods like uninformed-students [9], GT [18], and DSVDD [33] keep only the valuable information of normal data by building a compact latent feature space, in contrast to AE-based ones that try to miss the least amount of normal
Figure 2: Visualized summary of our proposed framework. A smaller cloner network, $C$, is trained to imitate the whole behavior of a source network, $S$ (VGG-16), on normal data. The discrepancy of their intermediate behavior is formulated by a total loss function and is used to detect anomalies at the test time. A hypothetical example of distance vectors between the activations of $C$ and $S$ on anomalous and normal data is also depicted. Interpretability algorithms are employed to yield pixel-precise anomaly localization maps.

3. Method

3.1. Our Approach

Given a training dataset $D_{train} = \{x_1, ..., x_n\}$ consisting only of normal images (i.e., no anomalies in them), we aim to train a cloner network, $C$, that detects anomalous images in the test set, $D_{test}$, and localizes anomalies in those images with the help of a pre-trained network. As $C$ needs to predict each sample’s deviation from the manifold of normal data, it needs to know the manifold quite well. Therefore, it is trained to mimic the comprehensive behavior of an expert network, called the source network $S$. Earlier works in knowledge distillation have conducted tremendous efforts to transfer one network knowledge to another smaller one to save computational cost and memory usage. Many of them strive to teach just the output of $S$ to $C$. We, however, aim to transfer the intermediate knowledge of $S$ on the normal training data to $C$ as well.

In [32], it is shown that by using a single intermediate-level hint from the source, a thinner but deeper cloner even outperforms the source on classification tasks. In this work, we provide $C$ with multiple intermediate hints from $S$ by encouraging $C$ to learn $S$’s knowledge on normal samples through conforming its intermediate representations in several critical layers to $S$’s representations. It is known that layers of neural networks correspond to features at various abstraction levels. For instance, first layer filters act as simple edge detectors. They represent more semantic features when considering later layers. Therefore, mimicking different layers educates $C$ in various abstraction levels, which
leads to a more thorough final understanding of the normal data. In contrast, using only the last layer shares a small portion of S’s knowledge with C. Besides, this causes the optimization to stuck in irrelevant local minima, especially when dealing with subtle anomalies that share almost all semantic concepts. On the contrary, using several intermediate hints turns the ill-posed problem into a more well-posed one. The effect of considering different layers in our method is more investigated in Sec. 3.3.1.

In what follows, we refer to the i-th critical layer in the networks as CP_i (CP_1 stands for the raw input) and the source activation values of that critical layer as a^{CP}_i, and the cloner’s ones as a^{CP}_c. As discussed in the knowledge distillation literature [32, 50], the notion of knowledge can be seen as the value of activation functions. We define the notion of knowledge as both the value and direction of all a^{CP}_i’s and a^{CP}_c’s activation values at each CP_i. 

Thus, \( L_{val} \) is formulated as

\[
L_{val} = \sum_{i=1}^{N_{CP}} \frac{1}{N_i} \sum_{j=1}^{N_i} (a^{CP}_c(j) - a^{CP}_s(j))^2, \tag{1}
\]

where \( N_i \) indicates the number of neurons in layer \( CP_i \) and \( a^{CP}_i(j) \) is the value of \( j \)-th activation in layer \( CP_i \). \( N_{CP} \) represents the total number of critical layers.

Additionally, we use \( L_{dir} \) to increase the directional similarity between the activation vectors. This is more vital in ReLU networks whose neurons are activated only after exceeding a zero value threshold. This indicates that two activation vectors with the same Euclidean distance from the target vector, may have contrasting behaviors in activating a following neuron. For instance, for \( e \) being a positive number, let \( a_1 = (0, 0, e, 0, \ldots, 0) \in \mathbb{R}^k \), \( a_2 = (0, (\sqrt{2} + 1)e, 0, 0, \ldots, 0) \in \mathbb{R}^k \) be activation vectors of two disparate cloner networks both trying to mimic the activation vector of a source network, \( a^* \), defined as \( a^* = (0, e, 0, 0, \ldots, 0) \in \mathbb{R}^k \). It is clear \( a_1 \) and \( a_2 \) have the same Euclidean distance from \( a^* \). However, assuming \( W = (0, 1, \ldots, 0, 0) \) as the weight vector of a neuron in the next layer of the network, we have

\[
W^T a_1 = 0 \leq 0,
W^T a_2 = (\sqrt{2} + 1)e > 0,
W^T a^* = e > 0. \tag{2}
\]

This means that the corresponding ReLU neuron would be activated by \( a_2 \), similar to \( a^* \), while deactivated by \( a_1 \). To address this, using the cosine similarity metric, we define the \( L_{dir} \) as

\[
L_{dir} = \sum_i 1 - \frac{\text{vec}(a^{CP}_c)^T \cdot \text{vec}(a^{CP}_s)}{\|\text{vec}(a^{CP}_c)\| \|\text{vec}(a^{CP}_s)\|}, \tag{3}
\]

where \( \text{vec}(x) \) is a vectorization function transforming a matrix \( x \) with arbitrary dimensions into a 1-D vector. This encourages the activation vector of \( C \) be not only close to the \( S \)’s one in terms of Euclidean distance but also be in the same direction. Note that \( L_{dir} \) is 1 for \( a_1 \), and is 0 for \( a_2 \). The role of \( L_{dir} \) and \( L_{val} \) is more elaborated in Sec. 3.3.3. Using the two aforementioned losses, \( L_{total} \) is formulated as

\[
L_{total} = L_{val} + \lambda L_{dir}, \tag{4}
\]

where \( \lambda \) is set to make the scale of both constituent terms the same. For this, we find the initial amount of error for each term on the untrained network and set \( \lambda \) with respect to it. Training using \( L_{total} \), unlike many other methods [18, 9], continues to fully converge, which is the only accessible criterion to measure when to stop training epochs.

Moreover, the architecture of \( C \) is designed to be simpler than \( S \) to enable knowledge “distillation.” This compression of the network facilitates the concentration on normal main features. While the source needs to be a very deep and wide model to learn all necessary features to perform well on a large-scale domain dataset, like ImageNet [16], the goal of the cloner is simply acquiring the source’s knowledge of the normal data. Hence, superfluous filters are only detrimental by focusing on non-distinguishing features, present in normal and anomalous data. Compressing the source prevents such distractions for the model. This is more effective when the normal class’ boundary and abnormal samples are extremely close, e.g., MVTecAD screw class. We investigate this effect more in Sec. 3.3.2.

\textbf{Anomaly Detection:} To detect anomalous samples, each test input is fed to both \( S \) and \( C \). As \( S \) has only taught the normal point of view to \( C \), anomalies, inputs out of the normal manifold, are a potential surprise for \( C \). In contrast, \( S \) has insights about images out of the normal manifold as well. All this leads to a potential discrepancy in their behavior for anomalous inputs, which could be detected by thresholding the loss in Eq. 4.

\textbf{Anomaly Localization:} [15, 58] have shown that derivative of the loss function with respect to the input has meaningful information about the significance of each pixel for the loss value. Hence, we employ the gradients of \( L_{total} \) with respect to the input to find the most impactful pixels on the loss function, i.e., anomalous regions in anomalous samples. To obtain our localization map for an input \( x \), we first acquire the attribution map, \( \Lambda \), by

\[
\Lambda = \frac{\partial L_{total}}{\partial x}. \tag{5}
\]

To reduce the natural noises in these maps, we induce Gaussian blur and opening morphological filter on \( \Lambda \). Hence, the localization map, \( L_{map} \), is achieved by

\[
M = g_\sigma(\Lambda),
L_{map} = (M \ominus B) \oplus B, \tag{6}
\]

where \( g_\sigma \) is a Gaussian function with standard deviation \( \sigma \), and \( B \) is a structuring element.
where $g$ denotes the Gaussian filtering with the standard deviation of $\sigma$. $\ominus$ and $\oplus$ represent morphological erosion and dilation by a structuring element $B$, respectively. Together, called opening, these operations remove small sporadic noises and yield clean maps. The structuring element, $B$, is a simple binary map usually in the shape of an ellipse or disk. Instead of using simple gradients as in Eq. 5, some other gradient-based interpretability methods can be employed to further illuminate the role of each pixel on loss value. We discuss different methods more in Sec. 3.3.4. Our proposed framework is illustrated in Fig. 2. Note that we need only two forward passes for detection and one backward pass through $C$ for localization at the test time.

3.2. Settings

VGG [43] features have shown remarkable performance in classification and transfer learning [46, 48]. This highlights the practicality of its filters in different domains. By transferring the knowledge of an ImageNet VGG-16 to a simple cloner, we exploit the discrepancy of features between $C$ and $S$ to find anomalies. In our VGG-16 source network, we choose the four final layers of each convolutional block, i.e., max-pooling layers, to be the critical points ($CP_i$'s). Selecting critical points is explored more in Sec. 3.3.1. For the cloner network, we use the architecture described in Fig. 2, which is smaller than the source for all experiments and datasets. As a result, it can benefit from the advantages of compression discussed in Sec. 3. The role of cloner architecture is discussed more in Sec. 3.3.2. Note that, similar to [33], we avoid using bias terms in our cloner’s network. As proven by [33], networks with bias in any layer can easily learn constant functions, independent of the input. In our work, though it can be negligible on datasets with diverse normal data, it can be detrimental when normal images are roughly the same. To be more specific, for some layers $l$ and $l+1$ that are between any $i$-th and $(i-1)$-th $CP$, the cloner can generate a specific constant activation vector, $a_{CP_i}$, regardless of the input, only by setting the $l$-th layer weights to zero and adjusting the $l+1$-th layer bias. As the normal training images are much alike, the intermediate source activations are also similar for them. Therefore, those constant $a_{CP_i}$'s can be arbitrarily close to the source correlated intermediate activations for any training input, which is the goal of the training phase, while harming the test procedure since they are constant outputs indeed. To avoid this, we use a bias-less network for $C$. In all experiments, we use Adam optimizer [21] with learning rate $= 0.001$ and batch size $= 64$ for optimization.

3.3. Ablation Studies

3.3.1 Intermediate Knowledge

In this experiment, we examine the effect of involving the last, the last two, and the last four max-pooling layers as $CP_i$'s on MVTecAD and MNIST. We report the average AUROC of all classes in Fig. 3.3.1. Clearly, a consistent growth trend exists that shows the effectiveness of considering more layers. Notice that some MVTecAD classes (e.g. “screw”) have near-random AUROC in “just the last layer” setting. This suggests that using just the last layer makes the problem ill-posed and hard to optimize.

Figure 3: The performance of our proposed method using various layers for distillation. More intermediate layers lead to a performance boost on anomaly detection.

3.3.2 Distillation Effect (Compact $C$)

As motivated initially in the knowledge distillation field, smaller $C$ plays an essential role in our approach by eliminating non-distinguishing filters causing various distractions. It is especially more important when performing on normal data, where the scope is dramatically limited. Here, we probe the effect of the cloner architecture. As in Fig. 4, anomaly detection, on MVTecAD, using a compact $C$ outperforms a $C$ with equal size to $S$. This is especially noticeable in the classes where anomalies are partial (like in “toothbrush” or “screw”). Overall, the smaller network performs better with a margin of $\sim 3\%$.

Figure 4: The performance of our proposed method using different equal/smaller cloner architectures compared to the source. The smaller network performs better in general.
3.3.3 $L_{dir}$ and $L_{val}$

In this part, we discuss each loss component effect to show the insufficiency of solely considering the Euclidean distance or directional loss in practice. The high impact of using $L_{total}$ can be seen in Fig. 5. We report the mean AUROC over all classes in the datasets. Refer to the supplementary for a class-detailed report. Using only $L_{dir}$ shows top results in cases where anomalies are essentially different from normal cases and are more diverse, like in CIFAR-10. In contrast, in cases with subtle anomalies, as in MVTecAD, MSE loss performs better. While $L_{dir}$ and $L_{val}$ fail noticeably in either of the cases, our proposed $L_{total}$, which is a combination of the two losses, achieves the highest performance considering both classification-based and defect detection settings at the same time. These results highlight the positive impact of considering a direction-wise notion of knowledge in addition to an MSE approach.

![Figure 5: The performance of our proposed method using different loss functions. $L_{total}$ performs well on both cases while individual directional or Euclidean losses fail in one.](image)

Table 1: Pixel-wise (AUROC) of anomaly localization on MVTecAD using different interpretability methods with and without Gaussian filtering.

<table>
<thead>
<tr>
<th>Method</th>
<th>Gradients</th>
<th>SmoothGrad</th>
<th>GBP</th>
</tr>
</thead>
<tbody>
<tr>
<td>Without Gaussian Filter</td>
<td>86.16%</td>
<td>86.97%</td>
<td>84.38%</td>
</tr>
<tr>
<td>With Gaussian Filter</td>
<td>90.51%</td>
<td>90.54%</td>
<td>90.08%</td>
</tr>
</tbody>
</table>

4. Experiments

In this section, extensive experiments have been done to demonstrate the effectiveness of our method. As explored in [39], some methods’ performance are harmed if trained for more than their hard-coded number of epochs. We report our results on average of the 10 last epochs for 10 different seeds plus the variances to show our stability. We report our method’s running time in the supplementary. We stress that $S$ is pre-trained on ImageNet and has not seen any data from the test datasets. Hence, the comparison is fair.

4.1. Experimental Setup

Datasets: We test our method on 7 datasets as follows: MNIST [24]: 60k training and 10k test 28 × 28 gray-scale handwritten digit images. Fashion-MNIST [49]: similar to MNIST (with 10k more training images) made up of 10 fashion product categories. CIFAR-10 [23] 50k training and 10k test 32 × 32 color images in 10 equally-sized natural entity classes. MVTecAD [7]: an industrial dataset with over 5k high-resolution images in 15 categories of objects and textures. Each category has both normal images and anomalous images having various kinds of defects (only for testing). All images have been down-scaled to the size 128 × 128. Retinal OCT Images (optical coherence tomography) [17]: consisting of 84,495 X-Ray images and 4 categories. HeadCT [22]: a medical dataset containing 100 128 × 128 normal head CT images and 100 with hemorrhage. Each image comes from a different person. BrainMRI for brain tumor detection [13]: consisting of 98 256 × 256 normal MRI images and 155 with tumors.

Evaluation Protocol: Medical datasets: 10 random normal images + all anomalous ones for test, the rest normal ones for training. MVTecAD & Retinal-OCT: datasets train and test sets are used. Others: one class as normal and others as anomaly, at testing: the whole test set is used.

4.2. Results

4.2.1 MNIST & Fashion-MNIST & CIFAR10

First, we evaluate our method on the conventional AD task on MNIST, Fashion-MNIST, and CIFAR-10 as described in Sec. 4.1. This targets detecting anomalies disparate from

[Code to reproduce the results is provided at https://github.com/rohban-lab/Knowledge_Distillation_AD](https://github.com/rohban-lab/Knowledge_Distillation_AD)
4.2.2 MVTecAD

**Detection:** In this part, we report the results of our method performance on AD using MVTecAD. As shown in Table 3, our method outperforms all others with a large margin of ~10%. This is remarkable since other methods fail to perform well in both one-class settings and defect detection simultaneously. In contrast, we achieve SOTA in both cases.

**Localization:** We not only accomplish SOTA in AD but we also outperform previous SOTA methods in anomaly localization. As stated in 3.3.4, we use simple gradients to obtain maps. We use the Gaussian filter with $\sigma = 4$ and a $3 \times 3$ ellipse structural element kernel. We compare our method against others in Table 4. We use AUROC, based on each pixel anomaly score, to measure how well anomalies are localized. Vividly, we outperform previous methods. Fig. 6 shows our localization maps on different defect types in MVTecAD.

### 4.2.3 Medical Datasets

To further evaluate our method in various domains, we use three medical datasets and compare our method with others.
shows that by utilizing augmentation detection using brain MRIs. In this dataset, images from different domains, even with domains far from the ImageNet domain. This shows that the knowledge of the pre-trained network, S, has been precious to the cloner, C, even in an entirely different domain of medical retinal OCT inputs. Furthermore, the unawareness of C about the outside of the normal data manifold intensifies the discrepancy between them. This expresses the generality of our method to even future unseen datasets, something missed in many methods.

Moreover, we validate our performance on brain tumor detection using brain MRIs. In this dataset, images with tumors are assumed as anomalies while healthy ones are considered normal. In Table 6, our method achieves SOTA results alongside LSA. While slightly (~ 0.5%) less than LSA, our method shows a significantly lower variance, magnifying its stability compared to the others. It is also noteworthy that LSA fails substantially on other tasks such as CIFAR-10 and MVTECAD anomaly detection with AUROCs ~ 23% and ~ 25% below our method’s, respectively. Lastly, using HeadCT (hemorrhage) dataset, we discuss an important aspect of our model. Performing on head computed tomography (CT) images for AD, we outperform OCGAN and GT by a considerable margin and perform ~ 3% below LSA. Since the training data is dramatically limited, our method may face difficulties transferring the S’s knowledge to C. However, this can be addressed by using simple data augmentations. We use 20-degree rotation in addition to scaling in the range of [0.9, 1.05] to augment the images. These augmentations are generic non-tuned ones aiming solely to increase the amount of data with no dependency on the dataset. Table 6 shows that by utilizing augmentations, the proposed method achieves similar results to LSA while outperforming it on other tasks significantly.

5. Conclusion

We show that “distilling” the intermediate knowledge of an ImageNet pre-trained expert network on anomaly-free data into a more compact cloner network, and then using their different behavior with different samples, sets a new direction for finding specific criteria to detect and localize anomalies. Without using intensive region-based training and testing, we leverage interpretability methods in our novel framework for obtaining localization maps. We achieve superior results in various tasks and on many datasets, even with domains far from the ImageNet domain.
References


[26] Zhe Li, Chong Wang, Mei Han, Yuan Xue, Wei Wei, Li-Jia Li, and Li Fei-Fei. Thoracic disease identification and localization with limited supervision. In Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition, pages 8290–8299, 2018.


