POLITECNICO
MILANO 1863

SCUOLA DI INGEGNERIA INDUSTRIALE
E DELL'INFORMAZIONE

Executive Summary of the Thesis

Leveraging Student-Teacher learning framework for semi-supervised visual anomaly detection and segmentation

Laurea Magistrale in Computer Science & Engineering - Ingegneria Informatica

Author: Alessandro Polidori

Advisor: Prof. Giacomo Boracchi

Academic year: 2021-2022

1. Introduction

Visual anomaly detection is a computer vision task which consists in detecting if an image is anomalous and, if possible, segmenting the anomalous regions. Traditional computer vision techniques have been widely used for industrial inspection to automatically detect defects since the birth of the research field. In recent years, similarly to what has happened over the past decade for the other visual recognition tasks like classification, object detection and segmentation, deep-learning based approaches started to achieve very competitive results in anomaly detection. In principle, AD could be framed as a multi-class classification problem, where the nominal (non-defective) is one of the classes.

In this work we will face the semi-supervised version of the problem: fitting a model using only nominal sample images. This is crucial in industrial settings, where it's unlikely to have at your disposal many occurrences of a defect. Many papers enriched the literature dedicated to this subject, especially in the last few years, providing a wide set of techniques.

We decided to focus on one of the most promising macro-category: techniques based on the Student-Teacher learning framework. These methods rely on the discrepancy between the feature maps generated by a powerful Teacher model and the corresponding ones produced by a weak Student (or an ensemble of them). This discrepancy is used as an anomaly score. These innovative methods already achieved remarkable results on MVTec AD datasets, which are a standard de facto for visual anomaly detection, but there is still much room for improvement.

In this thesis we investigate the two best known Student-Teacher based methods ([3],[4]) and we propose two novel solutions based on them. In the first one, inspired by [3], we add an anomaly scoring function based on the Students' ensemble variance to exploit the disagreement between the Students. In the second one we simplify Student's architecture in order to create a bottleneck and distill only essential knowledge. We show that both the solutions obtain promising results in terms of Image-level AUROC and Dice score on the MVTec AD datasets.

2. Problem Formulation

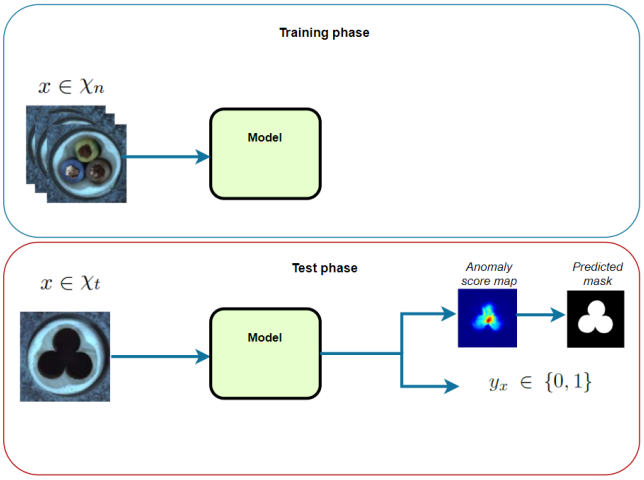

Figure 1: Inputs and outputs

Let us denote with $x \in \mathcal{X}_t$ a test image to be analyzed, defined over the pixel domain $\mathbb{Z}_p \in \mathbb{Z}^2$, with input size (h, w) and c channels. Given a channel, a pixel's intensity can range from 0 to $2^d - 1$, where d is the color depth. Visual anomaly detection can be defined as follows: given a test input image x, labeling it as anomalous (1) or normal (0) and, if anomalous, estimating the unknown anomaly mask $\Omega_x \in \mathbb{Z}_p \rightarrow \{0, 1\}$:

$$\Omega_x(i,j) = \begin{cases} 0 & \text{if pixel (i,j) is normal} \\ 1 & \text{if pixel (i,j) is anomalous} \end{cases}$$

Where Ω_x has size (h, w), equal to input image. So we will have two outputs: the image-level prediction $y_x \in \{0, 1\}$ and the predicted mask $\hat{\Omega}_x \in \mathbb{Z}_p \rightarrow \{0, 1\}$.

Given that we are considering a semi-supervised anomaly detection task, we assume that only images $x \in \mathcal{X}_n$ are used during training, where $\mathcal{X}_n$ is the set of non-defective (i.e. anomaly free) images. In our case, $\hat{\Omega}_x$ and y_x are obtained after choosing a threshold value. Our trained model $\mathcal{M}$ assigns an anomaly score to each pixel and an image-level score, which will then be converted to $\hat{\Omega}_x$ and y_x applying the threshold. Threshold can be chosen based, for example, on image-level F1-score maximization.

3. Background

3.1. Knowledge Distillation

Knowledge distillation was proposed as a compression technique for neural networks. It usually involves a knowledge transfer from a bigger pretrained network (Teacher) to an untrained lighter one (Student). Teacher and Student networks could have different architectures. The main objective is to ensure that the Student model imitates the Teacher. There are several possible approaches, in the next section we'll see the one exploited by our proposed solutions.

3.1.1 Feature-based distillation

An effective distillation strategy consist in transferring knowledge from the intermediate layers, to take into account multiple scales of feature representations. For every layer k we can establish a loss function:

$$\mathcal{L}_k = \|(f_t(\Phi_{t_k}(x)), f_s(\Phi_{s_k}(x)))\|_n \qquad (1)$$

where Φ_{t_k} and Φ_{s_k} are the k-th layer's feature maps of Teacher and Student networks respectively, f_t and f_s are the transformations needed in case they have different shapes and $\|\cdot\|_n$ is a distance. Note that other similarity functions can be used (e.g. cosine similarity).

3.2. Uninformed Students

Uninformed Students [3] is a semi-supervised technique that leverages the Student-Teacher framework to perform anomaly detection and segmentation. In this work the $(h \times w)$ images are fed in a patchwise manner. Meaning that for every pixel (m, n) of input image x, the sorrounding patch p is fed to the network. Firstly, a simple Teacher is trained on a large set of natural images (ImageNet) to match a pretrained network's output (last layer's feature maps):

$$\mathcal{L}_{teacher} = \|D(T(p)) - P(p)\|^2_{\ell 2} \qquad (2)$$

where D is a fully connected layer, referred as "decoder" in the paper, that is added to match the output dimension $(\mathbb{R}^d)$ of T with the pretrained network P.
Then, an ensemble of Students (same architecture as the Teacher) is trained to match the Teacher's output on nominal images of the target dataset:

$$\mathcal{L}_s = \frac{1}{wh} \sum_{(m,n)} \|\mu^s_{(m,n)} - (y^T_{(m,n)} - \mu) diag(\sigma^{-1})\|^2_{\ell 2} \tag{3}$$

where $\mu^s_{(m,n)}$ is the prediction made by the Student s for the pixel (m,n), $y^T_{(m,n)}$ is the Teacher's descriptor vector, while $\mu \in \mathbb{R}^d$ and $\sigma \in \mathbb{R}^d$ are the component-wise means and standard-deviations vectors computed on a set of non-defective validation images. The loss $\mathcal{L}_s$ is the scaled sum of all the pixels' prediction errors. The aim is to teach the Students to regress the output of the Teacher only on nominal patches. At test time, to evaluate the anomaly score in (m,n), the authors consider the regression error and, cleverly, the predictive uncertainty of the Students ensemble:

$$u_{(m,n)} = \frac{1}{M} \sum_{s=1}^{M} \|\mu^s_{(m,n)}\|^2_{\ell 2} - \|\mu_{(m,n)}\|^2_{\ell 2} \tag{4}$$

where M is the number of Students in the ensemble and $\mu_{(m,n)}$ is the ensemble's mean prediction. The assumption is that the group will give closer outputs on nominal patches. To manage multi-scale features, different networks with different patch-sizes are trained.

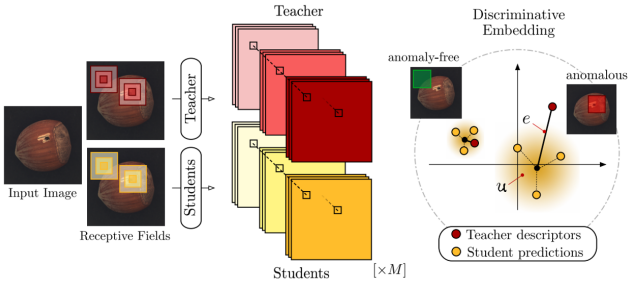

Figure 2: Uninformed Students schematic overview. During inference, ensemble's mean will yield a prediction error e and predictive uncertainty u ([3])

3.3. Student Teacher Feature Pyramid Matching

In [4], the authors solve the most critical issues of Uninformed Students and simplify the whole process at the same time. The first training phase gets bypassed using a pretrained ResNet as the Teacher network. A feature based distillation from multiple middle-layers (feature pyramid) removes the need for different patch-sizes, so a single network is capable to handle different scales of anomalies. Also, a single Student is used. Patch-wise processing is replaced by directly feeding the input image to the network. The Student is trained on nominal images to match Teacher's feature vectors setting as a distance metric the cosine similarity:

$$\mathcal{L}_k(i,j) = \frac{\|\hat{\Phi}_{t_k}(x)_{i,j} - \hat{\Phi}_{s_k}(x)_{i,j}\|^2_{\ell 2}}{2} \tag{5}$$

where $\Phi_{t_k}(x)_{i,j}$ and $\Phi_{s_k}(x)_{i,j}$ are the Teacher's and Student's k-th layer's feature vectors at position (i,j) in the feature maps. $\hat{\Phi}_{t_k}$ and $\hat{\Phi}_{s_k}$ are the ℓ_2 normalized versions. Note that the cosine similarity (5) can be expressed in terms of ℓ_2 distance when the vectors are normalized.

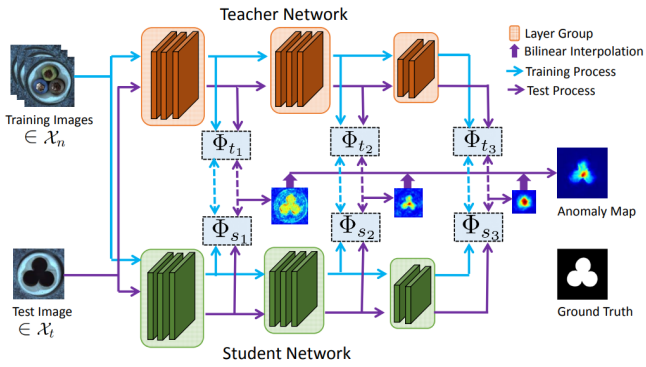

Figure 3: STFPM feature-based distillation ([4])

4. Proposed Solutions

4.1. Overview

Uninformed Students and STFPM both rely on the Student-Teacher framework and achieve remarkable results on the MVTec Anomaly Detection Datasets [2], a benchmark that became the de facto standard for semi-supervised visual anomaly detection in recent years. In this work we propose two new alternative methods based on these approaches. Given the simplified training scheme, we decided to use STFPM as a reference to build our methods. So the solutions introduced by STFPM and described in the previous chapter such as the multi-layer feature-based distillation and the whole-image processing will be employed in our proposed methodologies.

4.2. Solution 1: Feature Vectors Variance

In the first method, inspired by ([3]), we exploit an ensemble of Students. The aim is to equally train them on nominal images and then, at test time, use the variance of their feature vectors as an additional anomaly scoring function. The underlying assumption is that the ensemble's feature vectors will be less similar in correspondence of anomalous image regions. At first we separately train each Student to regress the Teacher's intermediate feature maps on anomaly-free images. As loss we employ the cosine similarity (5).

4.2.1 Students Uncertainty Measure

In Uninformed Students, a descriptor vector is obtained for each image patch (one patch for each pixel). The ensemble's predictive uncertainty with respect to these vectors' elements is measured for each pixel by (4). In our case, instead, we design a new uncertainty measure. To obtain it we first ℓ_2-normalize the Students' feature vectors:

$$\hat{\Phi}_{s_k}(x)_{i,j} = \frac{\Phi_{s_k}(x)_{i,j}}{\|\Phi_{s_k}(x)_{i,j}\|_{\ell 2}} \tag{6}$$

Where, given the student s, a vector $\Phi_{s_k}(x)_{i,j}$ is the concatenation of elements in position (i, j) along the k-th layer's feature maps. We need to measure the variance of these vectors' elements among the Students.
For each vector's element we compute standard-deviation amidst the ensemble, building a vector of the same length containing standard_deviation values and we denote it as $standard\_deviation_k(i, j)$.
The scalar uncertainty measure v_k is then:

$$v_k(i, j) = \|standard\_deviation_k(i, j)\|^2_{\ell 2} \tag{7}$$

4.2.2 Anomaly scoring function

To get the final anomaly score, we also need to compute the regression error between the ensemble's mean and the Teacher:

$$e_k(i, j) = \frac{\|\hat{\Phi}_{t_k}(x)_{i,j} - \overline{\hat{\Phi}}_{s_k}(x)_{i,j}\|^2_{\ell 2}}{2} \tag{8}$$

Where $\overline{\hat{\Phi}}_{s_k}(x)_{i,j}$ contains the average values of Students' (i, j) feature vectors.

Every k-th layer will be associated to an anomaly score map. This map will contain the sum between e_k and v_k in each (i, j) position.

4.3. Solution 2: Simple Student

The second method we propose is a variation of STFPM in which we simplify Student's architecture. That is not uncommon in a knowledge distillation setting, as the compression of the Student is usually the primary target. In STFPM, though, Student and Teacher models share the same architecture, since the authors are only interested in the anomaly detection task. What we argue is that the use of a simpler Student should increase the discrepancy between feature maps at test time when samples are anomalous. The rationale is that since Teacher's knowledge is distilled into a smaller architecture (less filters in convolutional layers), the distillation gets hindered. Therefore, Student network should learn to only represent essential nominal features needed to correctly reconstruct the feature maps.

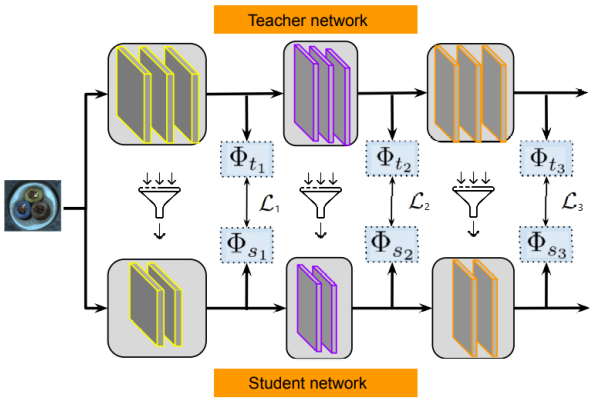

Figure 4: Simple Student feature-based distillation

In our implementation, we applied feature based distillation to layers 1,2,3 of a ResNet18. To get a more compact feature representation in each layer's latent space, we reduced the number of filters where possible. Note that, given that we are using ResNets, residual connections impose to have the same dimensions for input and output volumes in each basic block. Moreover, in layers 1,2,3 we also need to match Teacher's out-

puts dimensions (64, 128, 256) due to the feature based distillation.

5. Implementation Details

5.1. Anomaly maps upsampling

Note that, as in other approaches exploiting middle layers' activation maps, in our methods anomaly score maps need to be upsampled. That's a consequence of CNNs' pooling layers (e.g. maxpooling). Spatially reduced feature maps will produce anomaly score maps that are downsized with respect to input dimensions (h, w). Usually bilinear interpolation is applied to each k-th layer's map in order to reach input dimensions. After that, the resulting maps are combined through element-wise product to get scores that take into account different scales.

5.2. Setting Detection Threshold

In this work we are mostly interested in the anomaly detection aspect so we chose the threshold value that maximized the F1 score of the anomalous/nominal predictions $y_x \in \{0, 1\}$, disregarding the segmentation accuracy (for what concerns this choice). We always consider the image-level anomaly score as the maximum value of the combined anomaly score map described previously. Image-level F1 score optimization is not the only viable strategy. Another sensible choice would be to maximize the pixel-level F1 score, but that would imply that ground-truth masks are provided in the validation set, and that's not so common in industrial settings.

6. Experimental Results

In this section we show the results of our proposed methods and compare them with STFPM to appreciate the improvements brought by our contributions. All experiments were run on an Nvidia DGX system mounting Tesla V100 GPUs.

Figure 5: Example of segmentation output

6.1. Datasets

Tests have been executed on the MVTec AD datasets [2], a widespread benchmark composed of high resolution images divided into different subject categories (hazelnuts, bolts, carpets, pills...). Each of these categories contains a set of defect-free images to train the model. For testing, collections of defective images with hundreds of samples for every class of defect are included.

6.2. Metrics and Hyperparameters

In our tests we rely mainly on two metrics. AUROC (Area Under the Receiver Operating Characteristics) is very helpful for evaluating binary classification performance. In this case we are classifying anomalous (positive) and nominal (negative) samples. The ROC curve is plotted with True Positive Rate (TPR) on the y-axis against the False Positive Rate on the x-axis. The higher the area under the curve, the least amount of false positives are needed to obtain a certain TPR. Dice Score measures the overlap between segmentation and ground-truth mask. Even if we selected the threshold optimizing the image-level F1 score, it's still interesting to evaluate the pixel-level predictions of the model. Hyperparameters used in all the experiments are reported in the table below.

# Students	3
Batch Size	32
Input Size	224x224
Optimizer	SGD
Learning Rate	0.4
Momentum	0.9
Weight Decay	0.0001
feature extractor	ResNet18
Layer Blocks	1/2/3
Seed	42

6.3. Numerical Results

We report here the results obtained testing our methods on the MVtec AD Datasets. STFPM is used as a baseline. Best result in each category is bolded only for relevant improvements (> 0.005).

Image AUROC

	STFPM	Sol 1	Sol 2
Bottle	0.997	0.997	0.998
Cable	**0.962**	0.946	0.943
Capsule	0.450	**0.659**	0.544
Carpet	**0.977**	0.970	0.963
Hazelnut	0.974	0.971	**0.998**
Leather	1.0	1.0	0.997
Metalnut	0.954	**0.986**	0.558
Pill	0.516	**0.720**	0.448
Screw	0.748	**0.983**	0.956
Tile	0.983	**0.995**	0.988
Toothbrush	0.683	**0.763**	0.619
Transistor	**0.899**	0.874	0.814
Zipper	0.876	0.868	**0.912**
Wood	0.993	0.996	0.996

Dice Score

	STFPM	Sol 1	Sol 2
Bottle	0.661	0.663	0.555
Cable	0.513	**0.542**	0.488
Capsule	0.0133	**0.219**	0.0137
Carpet	0.641	0.642	0.594
Hazelnut	0.568	0.591	**0.610**
Leather	0.485	**0.507**	0.478
Metalnut	0.446	**0.605**	0.0104
Pill	0.0610	**0.132**	0.00157
Screw	0.120	**0.257**	0.238
Tile	0.520	**0.543**	0.519
Toothbrush	0.057	**0.126**	0.0643
Transistor	**0.611**	0.587	0.408
Zipper	0.446	0.448	0.417
Wood	0.573	0.559	0.570

7. Conclusions

For what concerns Feature Vectors Variance, the experimental results denote a clear improvement for many MVTec AD categories with respect to both Image AUROC and Dice Score. Notice how the Dice score improves in almost every category. A possible limitation concerns the computational resources needed by our Feature Vectors Variance solution. While STFPM only needs to keep two networks in memory, our technique requires an entire ensemble of Students, plus the Teacher. So the greater computational effort with respect to STFPM is undeniable. In some cases, though, a trade-off between resources and improved performances could be the most convenient choice. A trivial approach would be to use bigger networks like WideResNet-50, but that does not seem to improve the anomaly detection capabilities (see [1]). Therefore, in these situations our method offers a real advantage. Simple Student shows an improvement for less categories. That's likely a consequence of an excessive simplification of the Student's architecture. Still, results show that Simple Student can induce substantial improvements. A possible future development would be to better design the Student, maybe using 1×1 convolution in residual connections (to adapt basic block input/output tensors' sizes) and decoupling completely Student and Teacher architectures including fully connected layers to adapt the two networks' intermediate outputs dimensions (as it's done through the decoder in [3]). That would give more freedom to design a Student that is simpler but still able to represent nominal features well enough.

References

[1] Anomalib stfpm numerical results. `https://github.com/openvinotoolkit/anomalib/tree/main/anomalib/models/stfpm`. Accessed: 17-08-2022.

[2] Paul Bergmann, Kilian Batzner, Michael Fauser, David Sattlegger, and Carsten Steger. The mvtec anomaly detection dataset: A comprehensive real-world dataset for unsupervised anomaly detection. *International Journal of Computer Vision*, 129(4):1038–1059, Apr 2021.

[3] Paul Bergmann, Michael Fauser, David Sattlegger, and Carsten Steger. Uninformed students: Student-teacher anomaly detection with discriminative latent embeddings. *CoRR*, abs/1911.02357, 2019.

[4] Guodong Wang, Shumin Han, Errui Ding, and Di Huang. Student-teacher feature pyramid matching for anomaly detection. 06 2022.