

# Mixed supervision for surface-defect detection: from weakly to fully supervised learning

Jakob Božič, Domen Tabernik and Danijel Skočaj

University of Ljubljana, Faculty of Computer and Information Science

Večna pot 113, 1000 Ljubljana

[jakob.bozic@fri.uni-lj.si](mailto:jakob.bozic@fri.uni-lj.si), [domen.tabernik@fri.uni-lj.si](mailto:domen.tabernik@fri.uni-lj.si),

[danijel.skocaj@fri.uni-lj.si](mailto:danijel.skocaj@fri.uni-lj.si)

## Abstract

*Deep-learning methods have recently started being employed for addressing surface-defect detection problems in industrial quality control. However, with a large amount of data needed for learning, often requiring high-precision labels, many industrial problems cannot be easily solved, or the cost of the solutions would significantly increase due to the annotation requirements. In this work, we relax heavy requirements of fully supervised learning methods and reduce the need for highly detailed annotations. By proposing a deep-learning architecture, we explore the use of annotations of different details ranging from weak (image-level) labels through mixed supervision to full (pixel-level) annotations on the task of surface-defect detection. The proposed end-to-end architecture is composed of two sub-networks yielding defect segmentation and classification results. The proposed method is evaluated on several datasets for industrial quality inspection: KolektorSDD, DAGM and Severstal Steel Defect. We also present a new dataset termed KolektorSDD2 with over 3000 images containing several types of defects, obtained while addressing a real-world industrial problem. We demonstrate state-of-the-art results on all four datasets. The proposed method outperforms all related approaches in fully supervised settings and also outperforms weakly-supervised methods when only image-level labels are available. We also show that mixed supervision with only a handful of fully annotated samples added to weakly labelled training images can result in performance comparable to the fully supervised model's performance but at a significantly lower annotation cost.*

## 1. Introduction

Surface-quality inspection of production items is an important part of industrial production processes. Traditionally, classical machine-vision methods have been applied to

automate visual quality inspection processes, however, with the introduction of Industry 4.0 paradigm, deep-learning-based algorithms have started being employed [24, 41, 35, 42, 17, 40]. Large capacity for complex features and easy adaptation to different products and defects without explicit feature hand-engineering made deep-learning models well suited for industrial applications. However, an essential aspect of the deep-learning approaches is the need for a large amount of annotated data that can often be difficult to obtain in an industrial setting. In particular, the data acquisition is often constrained by the insufficient availability of the defective samples and limitations in the image labeling process. In this paper, we explore how these limitations can be addressed with weakly and fully supervised learning combined into a unified approach of mixed supervision for industrial surface-defect detection.

In large-scale production, the quantity of the items is not the main issue but the ratio between non-defective and defective samples is by design heavily skewed towards the defect-free items. Often, anomalous items are a hundred, a thousand, or even million-fold less likely to occur than the non-anomalous ones, leading to a limited set of defective items for training. Certain defect types can also be very rare with only a handful of samples available. This represents a major problem for supervised deep-learning methods that aim to learn specific characteristics of the defects from a large set of annotated samples.

The annotations themselves present an additional problem as they need to be detailed enough to differentiate the anomalous image regions from the regular ones properly. Pixel-level annotations are needed, but in practice, they are often difficult to produce. This is particularly problematic when it is challenging to explicitly define the boundaries between the defects and the regular surface appearance. Moreover, creating accurate pixel-level annotations is tiresome and costly to produce even when the boundaries are clear. Therefore, it is essential to minimize the labelling effort by

decreasing the required amount of annotations and reducing the expected labels' precision.

Various unsupervised [34, 4, 43] and weakly supervised [17, 44, 47] deep-learning approaches have been developed to reduce the need for a large amount of annotated data. The former are designed to train the models using defect-free images only, while the latter utilise weak labels and do not require pixel-level annotations. Although both approaches significantly reduce the cost of acquiring the annotated data, they significantly underperform in comparison with fully supervised methods on the defect detection task. Moreover, for many industrial problems, a small amount of fully annotated data is often available and can be used for training the models to improve the performance. This can result in a mixed supervision mode with some fully labeled samples and a number of weakly labeled ones as depicted in Fig. 1.

Unsupervised and weakly supervised methods are not able to utilize these available data. Mixed supervision has been applied on other computer vision tasks, such as image segmentation [33, 22], however, it has not yet been considered for industrial surface-defect detection.

In this work, we focus on a deep-learning approach suitable for industrial quality-control problems with varying availability of the defective samples and their annotations. In particular, we propose a deep-learning model for anomaly detection trained with weak supervision at the image-level labels, while at the same time utilising full supervision at the pixel-level labels when available. The proposed network can therefore work in a mixed supervision mode, fully exploiting the data and annotations available. By changing the amount and details of the required labels, the approach provides an option to find the most suitable trade-off between the annotation cost and classification performance. The mixed supervision is realised by implementing an end-to-end architecture with two sub-networks that are being simultaneously trained utilizing pixel-level annotations in the first sub-network and image-level annotations in the second sub-network. While pixel-level labels can be utilized during the training to further increase the performance of the proposed method, our primary goal is not to segment the defects, but rather to identify images containing defective surfaces. In addition, we also account for spatial uncertainty in coarse region-based annotations to further reduce the annotation cost.

We performed an extensive evaluation of the proposed approach on several industrial problems. We demonstrate the performance on DAGM [40], KolektorSDD [35] and Severstal Steel [1] datasets. Due to the lack of a real-world, unsaturated, large, and well-annotated surface-defect dataset, we also compiled and made publicly available a novel dataset termed *KolektorSDD2*. It is based on a practical, real-world example, consisting of over 3000 images

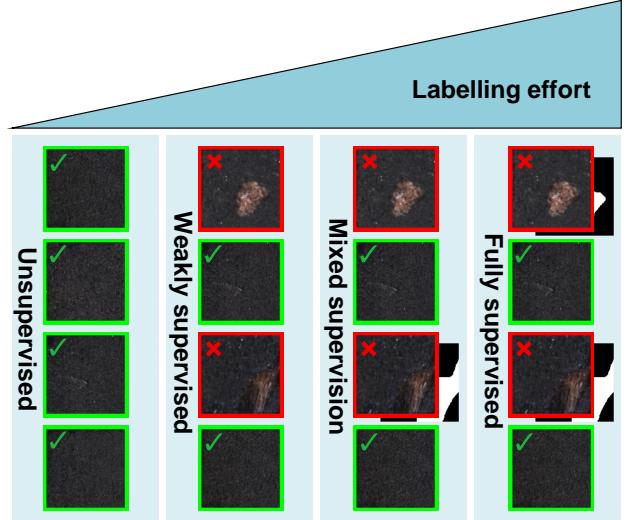


Figure 1: Visualization of several types of supervision and their required labelling effort.

containing several defect types of various difficulty. Images have been carefully annotated to facilitate accurate benchmarking of surface-defect-detection methods.

The proposed model outperforms related unsupervised and weakly supervised approaches when only weakly labeled annotations are available, while in the fully supervised scenario it also outperforms all other related defect-detection methods. Moreover, the proposed model demonstrates a significant performance improvement when only a few fully annotated images are added in mixed supervision. This often results in performance comparable to the results of fully supervised models but with a significantly reduced annotation cost.

The remainder of this paper is structured as follows: In Section 2, we present the related work, followed by the description of the proposed network with mixed supervision for surface-defect detection in Section 3. We present the details of the experimental setup and evaluation datasets in Section 4, while we present detailed evaluation results in Section 5. We conclude with a discussion in Section 6.

## 2. Related work

**Fully supervised defect detection** Several related works explored the use of deep-learning for industrial anomaly detection and categorisation [24, 18, 13, 42, 35, 27, 17, 38, 14], including the early work of Masci et al. [21] using a shallow network for steel defect classification, and a more comprehensive study of a modern deep network architecture by Weimer et al. [40]. From recent work, Kim et al. [14] used a VGG16 architecture pre-trained on general images for optical inspection of surfaces, while Wang

et al. [38] applied a custom 11-layer network for the same task. Rački et al. [27] further proposed to improve the efficiency of patch-based processing from [40] with a fully convolutional architecture and proposed a two-stage network architecture with a segmentation net for pixel-wise localization of the error and a classification network for per-image defect detection. In our more recent work [35], we performed an extensive study of the two-stage architecture with several additional improvements and showed the state-of-the-art results that outperformed others such as U-Net [28] and DeepLabv3 [6] on a real-world case of anomaly detection problem. We also extended this work and presented an end-to-end learning method for the two-stage approach [5], however still in a fully-supervised regime, without considering the task in the context of mixed or weakly supervised learning. Dong et al. [10] also used the U-Net architecture but combined it with SVM for classification and random forests for detection. Other recent approaches also explored lightweight networks [41, 18, 13, 20]. Lin et al. [18] used a compact multi-scale cascade CNN termed MobileNet-v2-dense for surface-defect detection while Huang et al. [13] proposed an even more lightweight network using atrous spatial pyramid pooling (ASPP) and depthwise separable convolution.

**Unsupervised learning** In unsupervised learning, annotations are not needed (and are not taken into account even when available) and features are learned from either reconstruction objective [15, 7], adversarial loss [12] or similar self-supervised objective [8, 39, 45]. In unsupervised anomaly detection solutions, the models are usually trained considering only non-anomalous image by applying out-of-distribution detection of anomalies as a significant deviations in features. Various methods based on this principle were proposed, such as AnoGAN [31] and its successor f-AnoGAN [30] that utilize Generative Adversarial Networks, or a deep-metric-learning-based approach with triplet loss that learns features of non-anomalous samples [34], or approach that transfers pre-trained discriminative latent embedding into a smaller network using knowledge transfer for out-of-distribution detection [4], termed Uninformed Students. The latter achieved state-of-the-art results in unsupervised anomaly detection on the MVTec dataset [3], which, however, only partially reflects the complexity of real-world industrial examples.

**Weakly supervised learning** Various weakly supervised deep-learning approaches have been developed in the context of semantic segmentation and object detection [26, 29, 2, 16, 37, 36]. In early applications, convolutional neural networks were trained with image-tags using Multiple Instance Learning (MIL) [26] or with constrained optimization as in Constrained CNN [25]. The approach by Seleh

et al. [29] further used dense conditional random fields to generate foreground/background masks that act as priors on an object, while Bearman et al. [2] used a single-pixel point label of object location instead of image-tags. Ge et al. [11] used a segmentation-aggregation framework learned from weakly annotated visual data and applied it to insulator detection on power transmission lines. Others utilized class activation maps (CAM) [46]. Zhu et al. [47] applied CAM for instance segmentation, while Diba et al. [9] simultaneously addressed image classification, object detection, and semantic segmentation, where CAM from image classification is used in a separate cascaded network to improve the last two tasks.

Class activation maps were also applied to anomaly detection. Lin et al. [17] addressed defect detection in LED chips using CAM from the AlexNet architecture [46] to localize the defects, but learning only on the image-level labels. Zhang et al. [44] extended CAM for defect localization with bounding box prediction in their proposed CADN model. Their model directly predicts the bounding boxes from category-aware heatmaps while also using knowledge distillation to reduce the complexity of the final inference model. However, both methods do not consider pixel-level labels in the learning process, thus failing to utilize this information when available.

**Mixed supervision** Several related approaches also considered learning with different precision of labels. Souly et al. [33] combined fully labeled segmentation masks with unlabeled images for pixel-wise semantic segmentation tasks. They train the model in adversarial manner by generating images with GAN and include any provided weak, image-level labels to the discriminator in GAN that further improves the semantic segmentation. Mlynarski et al. [22] addressed the problem of segmenting brain tumors from magnetic resonance images. They proposed to use fully segmented images and combine them with weakly annotated image-level information. They focus on the goal of segmenting brain tumor images, while our primary concern is image-level anomaly detection in the industrial surface-defect-detection domain. They also do not perform any analysis of different mixtures of the weakly and fully supervised learning, which is the central point of this paper.

### 3. Anomaly detection with mixed supervision

In this section, we present a deep-learning model that can be trained on a mixture of fully (pixel-level) and weakly (image-level) labeled data. Learning on weakly labeled data requires a model capable of utilizing segmentation ground-truth/masks when they are available, however, it also needs to utilize image-level, i.e. class-only, labels as well. The proposed model is based on our previous architecture with

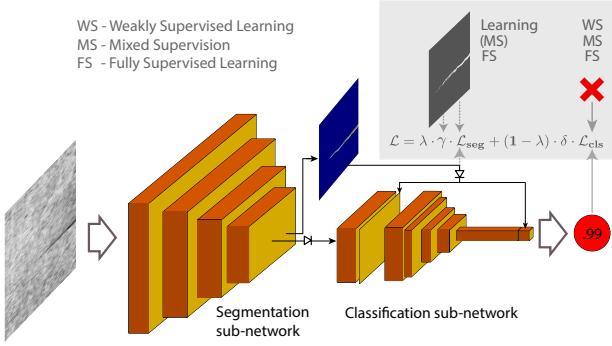


Figure 2: Proposed architecture with two sub-networks suitable for mixed supervision.

Segmentation sub-network			Classification sub-network		
Layer	Kernel size	Features	Layer	Kernel size	Features
<i>Input:</i> image		3/1	<i>Input:</i> $[S_f, S_h]$		1025
2x Conv2D	5x5	32	Max-pool	2x2	1025
Max-pool	2x2	32	Conv2D	5x5	8
3x Conv2D	5x5	64	Max-pool	2x2	8
Max-pool	2x2	64	Conv2D	5x5	16
4x Conv2D	5x5	64	Max-pool	2x2	16
Max-pool	2x2	64	Conv2D ( $C_f$ )	5x5	32
Conv2D ( $S_f$ )	5x5	1024	$[G_a(C_f), G_m(C_f), G_a(S_h), G_m(S_h)]$		66
Conv2D ( $S_h$ )	1x1	1	Fully connected ( $C_p$ )		1

Table 1: Architecture details for segmentation and classification sub-networks, in which *Features* column represent the number of output features.  $G_a$  and  $G_m$  represent global average and max pooling operations. Outputs of the network are a segmentation map  $S_h$  and a classification prediction  $C_p$ .

two sub-networks [35, 5], in which the segmentation sub-network utilizes fine pixel-level information and the classification sub-network utilizes coarse image-level information. The overall architecture with two sub-networks is depicted in Fig. 2, with the architectural details of each sub-network shown in Tab. 1.

### 3.1. Mixed supervision with end-to-end learning

Learning from a mixture of weakly labeled and fully labeled samples is only possible when both the classification and the segmentation sub-networks are trained simultaneously.

**End-to-end learning** We combine the segmentation and the classification losses into a single unified loss, which allows for a simultaneous learning in an end-to-end manner. The combined loss is defined as:

$$\mathcal{L}_{total} = \lambda \cdot \gamma \cdot \mathcal{L}_{seg} + (1 - \lambda) \cdot \delta \cdot \mathcal{L}_{cls}, \quad (1)$$

where  $\mathcal{L}_{seg}$  and  $\mathcal{L}_{cls}$  represent segmentation and classification losses, respectively. For both, we use the cross-entropy

loss. The remaining parameters are:  $\delta$  as an additional classification loss weight,  $\gamma$  as an indicator of the presence of pixel-level annotation and  $\lambda$  as a balancing factor that balances the contribution of each sub-network in the final loss. Note that  $\lambda$ ,  $\gamma$  and  $\delta$  do not replace the learning rate  $\eta$  in SGD, but complement it.  $\delta$  enables us to balance the contributions of both losses, which can be on different scales since the segmentation loss is averaged over all pixels, most of which are non-anomalous.

We also address the problem of learning the classification network on the initial unstable segmentation features by learning only the segmentation at the beginning and gradually progress towards learning only the classification at the end. We formulate this by computing their balancing factors as a simple linear function:

$$\lambda = 1 - n/n_{ep}, \quad (2)$$

where  $n$  is the current-epoch index and  $n_{ep}$  is the number of all training epochs. Without the gradual balancing of both losses, the learning would in some cases result in exploding gradients. We term the process of gradually shifting the training focus from segmentation to classification as the *dynamically balanced loss*. Additionally, using lower  $\delta$  values can further reduce the issue of learning on the noisy segmentation features early on.

**Using weakly labeled data** The proposed end-to-end learning pipeline is designed to enable utilisation of the weakly labeled data alongside fully labeled one. Such adoption of mixed supervision allows us to take a full advantage of any pixel-level labels when available, which weakly and unsupervised methods are not capable of using. We use  $\gamma$  from Eq. 1 to control the learning of the segmentation sub-network based on the presence of pixel-level annotation:

$$\gamma = \begin{cases} 1 & \text{negative image,} \\ 1 & \text{pos. image with pixel-level label,} \\ 0 & \text{pos. image with no pixel-level label.} \end{cases} \quad (3)$$

We disable the segmentation learning only when the pixel-level label for an anomalous (positive) image is not available. For non-anomalous (negative) training samples, the segmentation output should always be zero for all pixels, therefore, segmentation learning can still be performed. This allows us to treat non-anomalous training samples as fully labeled samples and enables us to train the segmentation sub-network in the supervised mode even in the absence of the pixel-level annotations during weakly supervised learning in the case of defect-free images.

**Gradient-flow adjustments** We stop the gradient-flow from the classification layers through the segmentation sub-network, which is required to stabilize the learning and enables training in a single end-to-end manner. During the

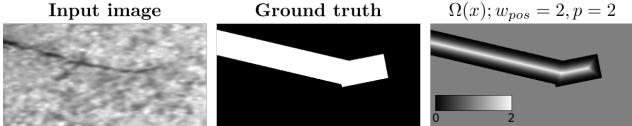


Figure 3: Segmentation loss weight mask obtained by applying distance transform algorithm on the label. Whiter shades on the segmentation loss mask indicate pixels with greater weight.

initial phases of the training, the segmentation sub-network does not yet produce meaningful outputs, hence, neither does the classification sub-network, therefore, error gradients back-propagating from the classification layers can negatively affect the segmentation part. We propose to completely stop those gradients, thus preventing the classification sub-network from changing the segmentation layers. We achieve this by stopping the error back-propagation at two points that connect segmentation and classification networks. The primary point is the use of segmentation features in the classification network. This is depicted by bottom diode symbol in Fig. 2. The second point, at which we stop error gradients is the max/avg-pooling shortcut used by the classification sub-network. This is depicted with the top diode symbol in Fig 2. Those shortcuts utilize the segmentation sub-network’s output map to speed-up the classification learning. Propagating gradients back through them would add error gradients to the segmentation’s output map, however, this is unnecessary due to the already available pixel-level ground-truth for those features. Without gradient-flow adjustments, we observed a drop in performance in fully supervised scenario and greater instabilities during the training in weak and mixed supervision.

### 3.2. Considering spatial label uncertainty

When only approximate, region-based labels are available, such as shown in Fig. 3, we propose to consider different pixels of the annotated defective regions differently. In particular, more attention is given to the center of the annotated regions and less to the outer parts. This alleviates the ambiguity arising at the edges of the defects where it is very uncertain whether the defect is present or not. This is implemented by weighting the segmentation loss accordingly. We weight the influence of each pixel at positive labels in accordance with its distance to the nearest negatively labelled pixel by using the distance transform algorithm.

We formulate weighting of the positive pixels as:

$$\mathcal{L}_{seg}(pix) = \Omega \left( \frac{\mathcal{D}(pix)}{\mathcal{D}(pix_{max})} \right) \cdot \hat{\mathcal{L}}(pix), \quad (4)$$

where  $\hat{\mathcal{L}}(pix)$  is the original loss of the pixel,  $\mathcal{D}(pix)/\mathcal{D}(pix_{max})$  is the distance to the nearest nega-

Dataset	Subset	# images	# classes	# defect types	Annotations
<b>DAGM</b>	<i>positive</i>	450	6	6	ellipse
	<i>negative</i>	3000			
<b>DAGM</b>	<i>positive</i>	600	4	4	ellipse
	<i>negative</i>	4000			
<b>KSDD</b>	<i>positive</i>	52	1	1	rotated bounding box
	<i>negative</i>	347			
<b>KSDD2</b>	<i>positive</i>	356	1	> 5	fine
	<i>negative</i>	2979			
<b>Severstal Steel</b>	<i>positive</i>	4759	1	> 5	fine or rotated bounding box
	<i>negative</i>	6666			

Table 2: Details of the evaluation datasets.

tive pixel normalized by the maximum distance value within the ground-truth region and  $\Omega(x)$  is a scaling function that converts the relative distance value into the weight for the loss. In general, the scaling function  $\Omega(x)$  can be defined differently depending on the defect and annotation type, however, we have found that a simple power function provides enough flexibility for different defect types:

$$\Omega(x) = w_{pos} \cdot x^p, \quad (5)$$

where  $p$  controls the rate of decreasing the pixel importance as it gets further away from the center, while  $w_{pos}$  is an additional scalar weight for all positive pixels. We have often found  $p = 1$  or  $p = 2$  as best performing, depending on the annotation type. Examples of a segmentation mask and weights are depicted in Fig. 3. Note, that weights for negatively labeled pixels remain 1.

## 4. Experimental setup

In this section, we detail the datasets and the performance metrics used to evaluate the proposed method as well as provide additional implementation details.

### 4.1. Datasets

We performed an extensive evaluation of the proposed method on four benchmark datasets. The summary of individual datasets is shown in Tab. 2, while additional details are provided below. A couple of images from all four datasets are presented in Figs. 4, 5, 8a, 8b and 8c.

**DAGM** The DAGM [40] dataset is a well-known benchmark dataset for surface-defect detection. It contains grayscale images of ten different computer-generated surfaces and various defects, such as scratches or spots. Each surface is treated as a binary-classification problem. Initially, six classes were presented while four additional ones were introduced later; consequently, some related methods report results only on the first six classes, while others report on all ten of them.

**KolektorSDD** The KolektorSDD [35] dataset contains grayscale images of a real-world production item; many of them contain visible surface cracks. Due to the small sample size, the images are split into three folds as in [35], while final results are reported as an average of three-fold cross validation.

**KolektorSDD2** Since the above mentioned datasets have been practically solved, and there is a huge need for real-world, reliable, complex and well annotated surface-detection datasets that would enable a fair comparison between different approaches, we compiled a novel dataset Kolektor Surface-Defect Dataset 2, abbreviated as KolektorSDD2<sup>1</sup>. The dataset is constructed from color images of defective production items, captured with a visual inspection system, that were provided and partially annotated by our industrial partner Kolektor Group d.o.o. Images for the proposed dataset were captured in a controlled environment and are of similar size, approximately 230 pixels wide and 630 pixels high. The dataset is split into the train and the test subsets, with 2085 negative and 246 positive samples in the train, and 894 negative and 110 positive samples in the test subset. Defects are annotated with fine-grained segmentation masks and vary in shape, size and color, ranging from small scratches and minor spots to large surface imperfections. Several images from this dataset are shown in Fig. 4.

**Severstal Steel defect dataset** The Severstal Steel defect dataset [1] is significantly larger than the other three datasets, containing 12,568 grayscale images in 4 classes, with various kinds of defects. We use only a subset of the dataset in our evaluation. In particular, we use all negative images but consider only positive images with the most common defect class present in the image (class 3). The defects are very diverse in size, shape and appearance, ranging from scratches and dents to excess material. Although the dataset is fairly large and diverse, some defects are quite ambiguous and may not be properly annotated as observed in the third example in Fig. 8c.

## 4.2. Performance metrics

In all experiments, we focus on evaluating per-image classification metrics, which are the most relevant metrics in the industrial quality control, since they decide whether to keep or discard the inspected items. In particular, we mostly measure the performance in terms of the average precision (AP), which is calculated as the area under the precision-recall curve. For the DAGM dataset, we also report several other metrics that are used to report the results in related lit-

<sup>1</sup>The dataset is publicly available at <https://www.vicos.si/Downloads/KolektorSDD2>



Figure 4: Examples of training images from KolektorSDD2 dataset with pixel-wise annotations shown in red-overlay region.

erature, to enable a fair comparison of the proposed method with related approaches.

## 4.3. Implementation details

The proposed architecture is implemented<sup>2</sup> in PyTorch framework. In all experiments, the network is trained with stochastic gradient descent, with no momentum and with no weight decay. For all experiments, we disable the gradient flow from classification network to segmentation network and employ loss weighting for positive pixels. We enable dynamically balanced loss for all cases with mixed supervision and fully supervised learning, but disable it when only weak labels are used. Since pixel-level labels are never available in weakly supervised case, it has proven more useful to immediately start learning classification layers instead of delaying it with the gradual increase of  $\lambda$ . To account for the unbalanced datasets, we employ undersampling of negative samples, in each training epoch we sample a selection of negative samples of equal size as the positive subset. We also ensure that over the course of learning, all negative images are used approximately as often. We also apply morphological dilation to the annotations in datasets where they are narrow and small in order to enlarge the label and facilitate distance-transform weighting. This has often proven better than using narrow annotations, as small defects can be difficult to precisely annotate, and, for some types of defects, the boundary between the defect and the

<sup>2</sup>Implementation is available on github: <https://github.com/vicoslab/mixed-segdec-net-comind2021>

background can not be precisely defined. Furthermore, distance transform assigns very small weights to the pixels near the edge of defects, thus dilating the annotations increases the weights for those pixels.

#### 4.4. Training and inference time

The proposed method is capable of real-time detection, achieving 57 fps on  $512 \times 512$  images (DAGM), 23 fps on  $512 \times 1408$  images (KSDD), 85 fps on  $232 \times 640$  images (KSDD2) and 36 fps on  $1600 \times 256$  images (Severstal Steel), running on a single Nvidia RTX2080Ti GPU. The training time depends on the dataset size. In our experiments, training a model on DAGM took 35 minutes (7 or 10 classes), on Severstal Steel ( $N_{all} = 3000$ ) it took 460 minutes, while for KSDD2 and KSDD (for a single fold) it took 15 and 32 minutes respectively.

#### 4.5. Related methods

We compare our approach against several related methods that have reported their results on DAGM [34, 44, 14, 27, 40, 18, 38, 13, 20] and KolektorSDD [10, 35, 19] in the literature. Additionally, we evaluate two state-of-the-art unsupervised methods that we apply to all four datasets. Below, we provide more details on the implementation of the two unsupervised methods.

**Uninformed students** Since the code for Uninformed-students [4] is not publicly available, we implemented our own version, which attains results comparable to the ones reported in the literature. We additionally opted to use a per-image defect score for classification similar to [23], which slightly improved results compared to the original implementation. For each image, we performed a  $21 \times 21$  pixel average pooling with stride = 1, to obtain patch scores, and then calculated the final image score as  $s = \max(patch)/(1+\text{avg}(pixel))$ , i.e. we took maximum patch score divided by average pixel score plus one. This increased the AP in all four datasets.

**f-AnoGAN** We used publicly the available TensorFlow implementation of f-AnoGAN [32]. In all experiments, we extracted 500 patches per image for training, with each patch  $64 \times 64$  pixels in size. In inference, a patch size of  $64 \times 64$  pixels was extracted and classified for every pixel location, which resulted in a heatmap of the same size as the original image. For scoring, we used the distance between the reconstructed image and the original image as well as between the discriminator features of the reconstructed and the original image. A mean squared error was used to measure both distances. We also opted to use normalization of the scoring similar to [23], which slightly improved results on noisy outputs.

Method	Type	10 classes				6 classes	
		AP	AUC	F1	CA	CA	mAcc
f-AnoGAN [32]	US	19.5	57.5	27.8	79.7	81.7	54.6
Uninf. stud. [4]	US	66.8	86.4	67.1	84.3	79.3	78.5
Staar [34]	US	-	83.0	-	-	-	-
CADN-W18 [44]	WS	-	-	63.2	86.2	-	-
CADN-W18(KD)	WS	-	-	65.8	87.6	-	-
CADN-W32 [44]	WS	-	-	69.0	89.1	-	-
Ours (N=0)	WS	74.0	86.1	74.6	89.7	85.4	81.4
Ours (N=5)	MS	91.5	94.9	92.3	92.9	88.1	91.6
Ours (N=15)	MS	100	100	100	100	100	100
Ours (N=45)	MS	100	100	100	100	100	100
Ours (N=N <sub>all</sub> )	FS	100	100	100	100	100	100
Kim [14]	FS	-	-	-	-	-	99.9
Rački [27]	FS	-	99.6	-	99.6	99.2	99.4
Weimer [40]	FS	-	-	-	-	-	99.2
Lin [18]	FS	-	99.0	-	-	-	99.8
Wang [38]	FS	-	-	-	-	99.8	99.4
Huang [13]	FS	-	-	-	-	-	99.8
Liu [20] (N <sub>all</sub> =20%)	FS	-	-	-	-	99.9	-

Table 3: Comparison with related work on the DAGM dataset. For AP, AUC, F1-measure and classification accuracy (CA), the results are averaged over all 10 classes, whereas for (second) CA and mAcc=(TPR+TNR)/2, they are averaged only over the first 6 classes for comparison with the related work that report results only in terms of those metrics.

## 5. Evaluation results

In this section, we present the results of the extensive evaluation of the proposed method on several industrial quality-control problems. We first present the evaluation on the DAGM dataset for comparison to different related works, and then present results on KolektorSDD, KolektorSDD2 and Severstal steel, all of which represent practical surface-anomaly detection problems taken from real-world industrial cases. Finally, we present a detailed ablation study that analyses the performance of individual components in the proposed method.

We simulate and evaluate different levels of supervision by varying the number of positive (i.e., anomalous) training samples for which we have available pixel-wise segmentation masks (i.e, the number of segmented anomalous images  $N$  used):

- i) **weak supervision**, with only image-level labels for all images (i.e.,  $N = 0$ ),
- ii) **mixed supervision**, with image-level labels for all images but also pixel-level labels only for a subset of anomalous images (i.e.,  $0 < N < N_{all}$ ), and
- iii) **full supervision**, with image-level and pixel-level labels for all anomalous images (i.e.,  $N = N_{all}$ ).

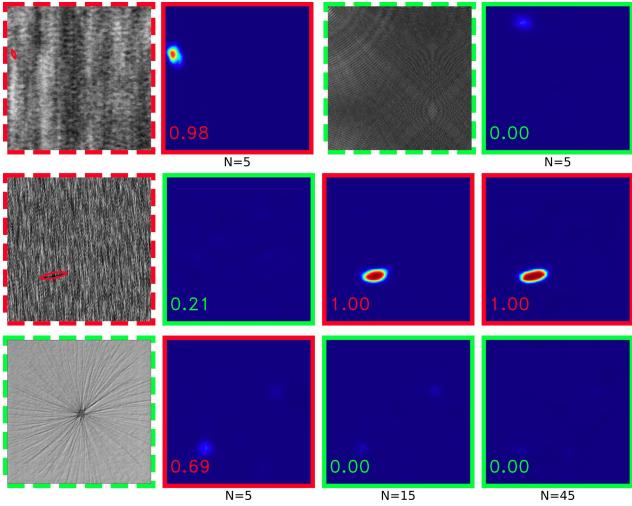


Figure 5: Examples of segmentation outputs and scores from the proposed method on the DAGM dataset. Number of fully labeled samples is marked with  $N$ . Border colour indicates presence (red) or absence (green) of defects, with dashed border around ground-truth images and solid border around predictions. The detection score that represents the probability of the defect is indicated in the bottom left corner.

Although we limit the number of images with the segmentation mask, we always use data with the image-level label, i.e. weak label that only indicates whether the anomaly is present in the sample or not.

## 5.1. DAGM

We first performed evaluation on the DAGM dataset. We consider the number of positive segmented samples  $N = \{0, 5, 15, 45, N_{all}\}$ , and use only image-level labels for the remaining training images. In all cases, we trained for  $n_{ep} = 70$  epochs, with the learning rate  $\eta = 0.05$ , batch size  $bs = 1$ ,  $\delta = 1$ ,  $w_{pos} = 10$  and  $p = 1$ . We dilated segmentation masks with  $7 \times 7$  kernel.

In Fig. 6, we show AP for all 10 classes of the dataset. For each class, we show AP of both unsupervised methods and of our model for all different values of  $N$ . On the bottom left part of each bar, we also indicate the percentage of the positive samples for which we used pixel-level labels,  $N/N_{all}$ , which are encoded as proportions of dark and light gray bars.

The proposed method achieves AP of over 90% for 6 classes even with  $N = 0$  and, on average, achieves 7.2 percentage points higher AP than the current state-of-the-art unsupervised approach. Introducing only five pixel-level labels allows the method to achieve 100% detection rate on 8 classes and raises the average AP to 91.5%. Finally, the

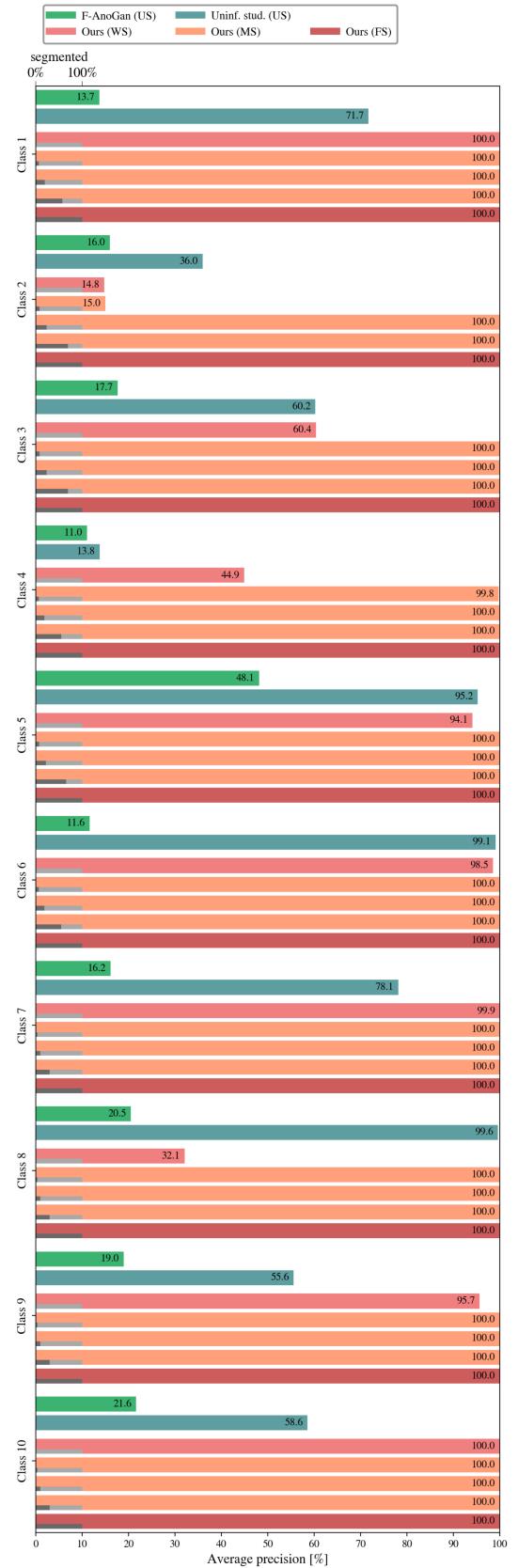


Figure 6: Results on the DAGM dataset in terms of average precision (AP).

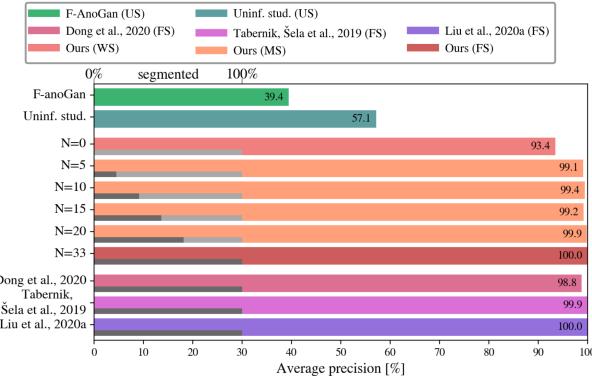


Figure 7: Results on the KolektorSDD dataset in terms of average precision (AP).

method achieves 100% detection rate on all 10 classes with only 15 pixel-level labels, thus outperforming other fully supervised approaches while having less than a quarter of positive samples with pixel-level labels. Several examples of detection are depicted in Fig. 5.

In Tab. 3, we compare the proposed method with a number of other approaches. We report CA and  $mAcc = TPR + TNR / 2$  metrics averaged over the first six classes since the related methods that use those metrics report results only on the first six classes. Other metrics are averaged over all 10 classes. The proposed method outperforms all other weakly supervised and unsupervised approaches shown at the top of the table and also all fully supervised methods shown at the bottom of the table. Moreover, 100% detection rate can be achieved in mixed supervision with only 15 fully annotated pixel-level samples. This points to superior performance and flexibility when allowing for mixed supervision.

## 5.2. KolektorSDD

On KolektorSDD dataset, we varied the number of segmented positive samples  $N = \{0, 5, 10, 15, 20, N_{all}\}$ , where  $N_{all} = 33$  and report the performance in terms of the average precision (AP). Identical hyper-parameters were used for all mixed and fully supervised runs, with  $n_{ep} = 50$ ,  $\eta = 1$ ,  $bs = 1$ ,  $\delta = 0.01$ ,  $w_{pos} = 1$  and  $p = 2$ , whereas we increased  $\delta$  to 1 and decreased  $\eta$  to 0.01 for run with no pixel-level annotations. We dilated segmentation masks with  $7 \times 7$  kernel.

In the Fig. 7, we show AP for various numbers of segmented positive samples. Without any segmented positive samples, our proposed approach already achieves AP of 93.4% and significantly outperforms the current state-of-the-art unsupervised methods that do not (and can not) use any pixel-wise information, by roughly doubling their results. When only 5 segmented samples are added, the

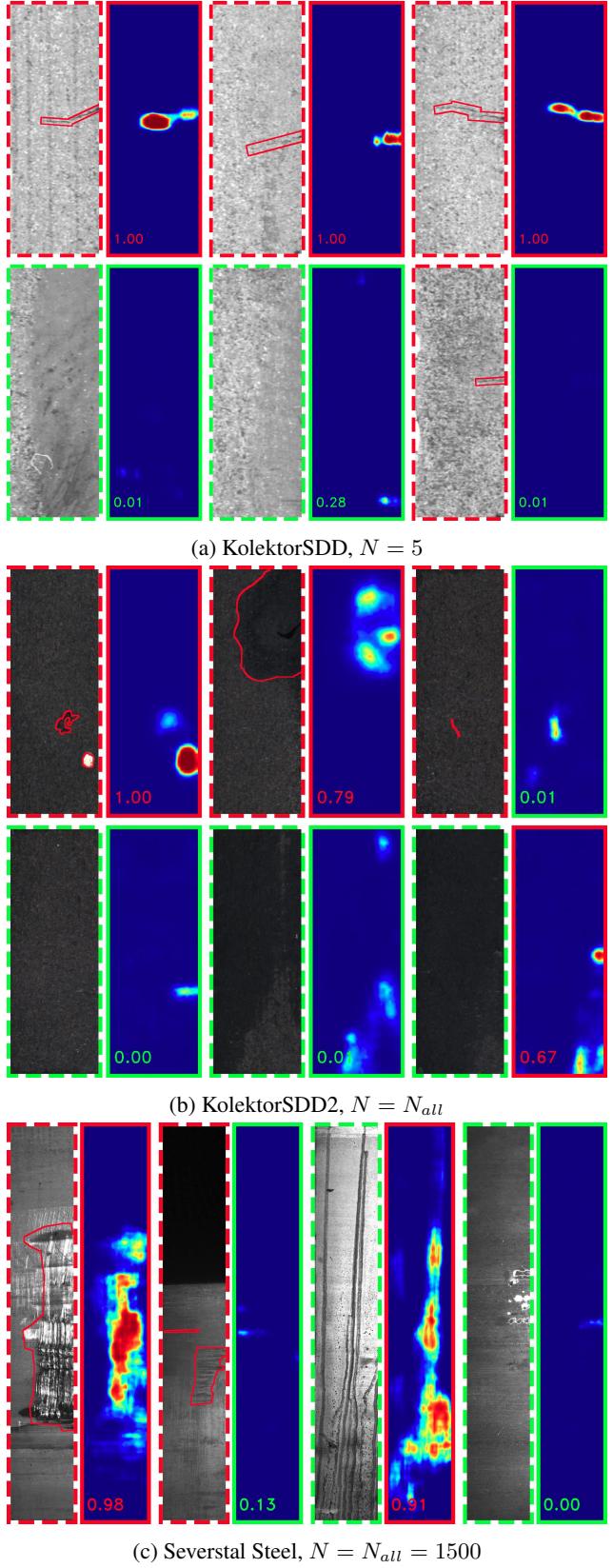


Figure 8: Examples of detections from (a) KolektorSDD, (b) KolektorSDD2 and (c) Severstal Steel defect dataset.

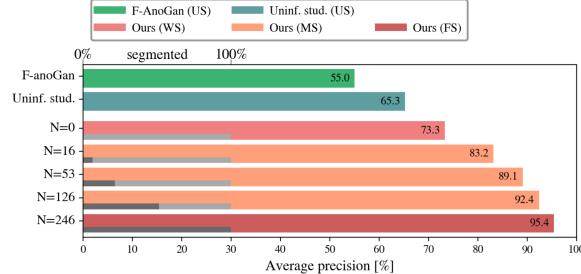


Figure 9: Results on the KolektorSDD2 dataset in terms of average precision (AP).

AP drastically increases to 99.1% and approaches fairly close to that of the fully supervised scenario. This demonstrates well that the full annotation is often not needed for all images and only a few annotated images are enough to achieve respectable results. In this case, less than 15% of images needed to be annotated, which can significantly reduce the annotation cost with almost no classification loss. When using all fully labeled samples the proposed approach achieves 100% detection rate. Several examples of defections are depicted in Fig. 8a.

### 5.3. KolektorSDD2

Next, we evaluate our approach on the newly proposed KolektorSDD2 dataset. We used identical hyper-parameters for all experiments,  $n_{ep} = 50$ ,  $\eta = 0.01$ ,  $bs = 1$ ,  $\delta = 1$ ,  $w_{pos} = 3$  and  $p = 2$ . Since the segmentation mask are very fine, we used a larger,  $15 \times 15$  kernel for dilation.

Results obtained on this challenging dataset are presented in Fig. 9, where AP for all values of  $N$  are shown. Even without any pixel-level labels, the proposed approach outperforms current state-of-the-art unsupervised methods and achieves AP of 73.3%. When 16 segmented samples are introduced, AP increases significantly to 83.2% and keeps increasing steadily with the introduction of additional pixel-level labels, reaching 95.4% when  $N = N_{all}$ . This demonstrates that the proposed method scales well with the introduction of segmented samples and is capable of taking the full advantage of all the information about the samples that it is presented with. Several examples of detections are depicted in Fig. 8b.

### 5.4. Severstal Steel defect dataset

Lastly, we evaluate the proposed method on Severstal Steel defect dataset. Since this is a large dataset, we also vary the overall number of positive training images available. We used  $N_{all} = \{300, 750, 1500, 3000\}$  positive samples, and then for each case used  $N = \{0, 10, 50, 150, 300\}$ , as well as  $N = \{750, 1500, 3000\}$ , where  $N \leq N_{all}$  that were segmented with pixel-wise ac-

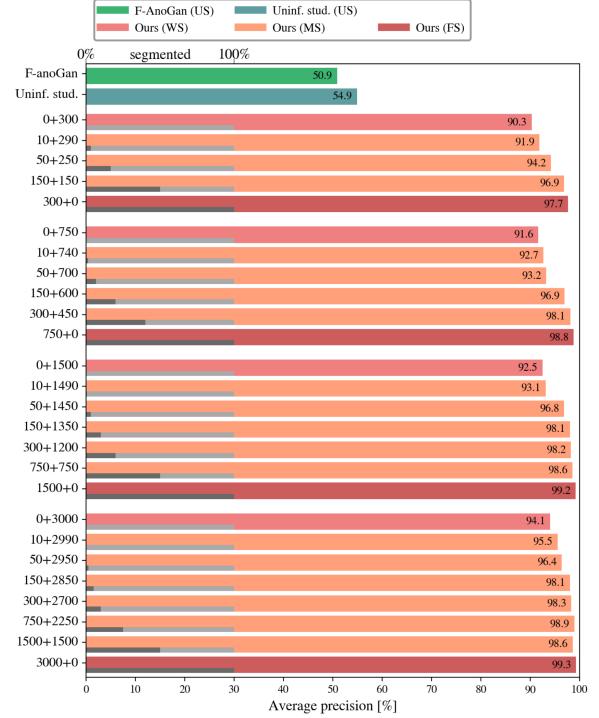


Figure 10: Results of evaluation on the Severstal steel defect database. The number of *fully + weakly* labeled positive samples are shown as labels to the left of the bars.

curacy. We used all negative samples regardless of used  $N$  and  $N_{all}$ . We trained for  $n_{ep} = \{90, 80, 60, 40\}$  when  $N_{all} = \{300, 750, 1500, 3000\}$ , respectively, and used  $\eta = 0.1$ ,  $bs = 10$ ,  $\delta = 0.1$ ,  $w_{pos} = 1$  and  $p = 2$ .

Fig. 10, shows AP for all  $N_{all}$  and  $N$ . Bars are grouped according to the  $N_{all}$ , with the labels showing the number of segmented positive samples and the number of positive samples without pixel-level labels. We observe that the proposed model is capable of learning from only image-level labeled samples, where it achieves AP of 90.3%, 91.6%, 92.5% and 94.1% for  $N = 0$  and  $N_{all} = \{300, 750, 1500, 3000\}$ . Note that with the positive set that is 10 times larger, the AP only increases for 3.8 percentage points (from 90.3% to 94.1%), pointing to a logarithmic increase in performance with the increase of the number of the positive samples. With the introduction of the pixel-level labels, the overall performance increases significantly, particularly for experiments with smaller  $N_{all}$ .

In the case of  $N_{all} = 300$ , the AP increases from 90.3% to 94.2% when only 50 pixel-wise labels were added and surpasses the AP achieved when using 3000 positive samples with image-level labels only. Furthermore, we can obtain similar results by using 3000 positive samples with only 150 pixel-level labels as with using 300 positive samples

with pixel-level labels. Several examples of detections are depicted in Fig. 8c.

## 5.5. Ablation study

Finally, we evaluate the impact of the individual components, namely the *dynamically balanced loss*, the *gradient-flow adjustment* and the *distance transform*, in the proposed model. We perform the ablation study on the DAGM, KolektorSDD, and Severstal Steel Defect datasets, however, for the later we use only a subset of images to reduce the training time, we used 1,000 positive and negative samples for training and the same amount for testing, we trained for 50 epochs.

We report the performance by gradually enabling individual components, and by disabling a specific component while leaving all the remaining ones enabled. Results are reported in Tab. 4. The results indicate that on all three datasets the worst performance is achieved without any component enabled while the best performance is achieved with all three components used. In the latter case, the proposed model is able to completely solve the DAGM and KolektorSDD datasets while achieving AP of 98.74% on Severstal Steel dataset. The second part of Tab. 4 also demonstrates the importance of each component for weakly supervised learning and for mixed supervision. In both cases, the best result on all three datasets is achieved only with all components enabled. Note that for weakly supervised learning, we did not have any pixel-level annotations and, therefore, could not apply the distance transform. We describe the contribution of each component to the overall improvements in more details below.

**Dynamically balanced loss** Enabling only dynamic balancing of losses with gradual inclusion of classification network already provides a boost of the performance in all three datasets. In fully supervised learning, the dynamically balanced loss improves AP by 1.08 percentage points in DAGM (from 89.02% to 90.10%), by 0.23 points in KolektorSDD (from 99.77% to 100.00%) and by 1.68 points in Severstal Steel (95.36% to 97.04%). Similar improvements can also be observed in mixed supervision, with 4.44, 0.75 and 1.99 percentage points improvements for DAGM, KolektorSDD and Severstal Steel.

**Gradient-flow adjustment** The gradient-flow adjustment has proven to be equally important as the dynamically balanced loss. Both improvements naturally result in a similar performance in KolektorSDD and Severstal Steel since they both prevent unstable segmentation features to significantly affect learning of the classification layers in the early stages. However, enabling both improvements is more robust as it eliminates the convergence issues while also improving results on all three datasets, especially for DAGM, in which

100% detection rate can be achieved for fully supervised learning and AP of 95.37% for mixed supervision. On the other hand, it has also proven better to use only gradient-flow adjustment in weak supervision and to completely disable dynamically balanced loss, since in this case segmentation loss is not present.

**Spatial label uncertainty** Lastly, enabling the distance transform as weights for the positive pixels pushes performance on KolektorSDD to 100% detection rate, therefore completely solving KolektorSDD and DAGM in fully supervised mode. It also improves AP on Severstal Steel dataset from 98.24% to 98.74%. Moreover, distance transform is even more important in mixed supervision where it enables 100% detection rate with only 25% of fully annotated data and using only weak labels for the remaining data.

## 6. Conclusion

In this paper, we presented a deep-learning model for surface-anomaly detection using mixed supervision. We proposed an architecture with two sub-networks: the segmentation sub-network that learns from pixel-level labels and the classification sub-network that learns from weak, image-level labels. Combining both sub-networks allowed for mixing of fully and weakly labeled data to achieve the best result with minimal annotation effort. To accomplish this, we proposed a unified network with end-to-end learning and enabled handling of coarse pixel-level labels. We demonstrated that the proposed model outperforms the existing state-of-the-art anomaly-detection models in both fully supervised mode and in weakly supervised mode. The network can be applied in either weakly supervised or fully supervised settings, or in a mixture of both, depending on the availability of annotations. The mixture of both annotation types has resulted in performance competitive to fully supervised mode while requiring significantly fewer and less complex annotations. We have demonstrated this on three existing industrial dataset: DAGM, KolektorSDD and Severstal Steel Defect, and have additionally proposed KolektorSDD2 that represents a real-world industrial problem with several defect types spread over 3000 images.

Two important conclusions can be drawn from the results. First, using a large number of only weakly labeled data has proven to be more important than using a 10-fold smaller set of fully labelled samples. In cases, such as existing industrial production control where weakly labeled data can be obtained at no cost, it is much less costly to use a large set of weakly labeled data without sacrificing the performance. However, as a second conclusion, we observed that a significant performance boost is often obtained when adding just 5–10% of fully-labeled data, resulting in almost the same performance as using all fully-labeled data. This

	DAGM		KolektorSDD		Severstal Steel		Dynamically balanced loss	Gradient-flow adjustment	Distance transform
	AP	FP+FN	AP	FP+FN	AP	FP+FN			
<i>FS (N=100%)</i>	89.02	1010+45	99.77	0+1	95.36	108+121			
	90.10	995+14	100.00	0+0	97.04	95+76	✓		
	100.00	0+0	99.88	1+0	98.24	52+58	✓	✓	
	91.88	471+68	99.88	1+0	97.80	57+60	✓		✓
	99.95	2+2	99.75	1+0	98.32	73+64		✓	✓
	100.00	0+0	100.00	0+0	98.74	59+40	✓	✓	✓
<i>MS (N ≈ 25%)</i>	66.02	2497+56	99.06	2+1	94.36	136+120			
	70.46	1775+65	99.81	2+0	96.35	113+96	✓		
	95.37	22+56	99.10	0+2	96.88	57+105	✓	✓	
	91.44	837+19	99.40	1+1	92.49	183+137	✓		✓
	99.93	0+3	99.16	1+2	96.60	122+83		✓	✓
	100.00	0+0	99.10	0+2	97.73	67+64	✓	✓	✓
<i>WS (%)</i>	74.05	438+248	93.43	4+5	91.01	201+158		✓	N/A
	61.49	1947+127	93.16	3+5	90.88	247+120	✓	✓	N/A

Table 4: Performance of individual components on three datasets by gradually including each one for fully supervised (FS), mixed supervision (MS), and weakly supervised (WS) learning. We report average precision (AP) and the number of false positives (FP) and false negatives (FN).

realization can considerably reduce the time and cost of data annotation in many industrial applications.

## Acknowledgments

This work was in part supported by the ARRS research project J2-9433 (DIVID) and research programme P2-0214. We would also like to thank Kolektor Group d.o.o. for providing images and initial annotations used in KolektorSDD2 dataset.

## References

- [1] Severstal: Steel Defect Detection on Kaggle Challenge. 2, 6
- [2] A. Bearman, V. Ferrari, and O. Russakovsky. What’s the point : Semantic segmentation with point supervision. In *ECCV*, pages 1–11, 2016. 3
- [3] P. Bergmann, M. Fauser, D. Sattlegger, and C. Steger. MVTec AD — A Comprehensive Real-World Dataset for Unsupervised Anomaly Detection. In *CVPR 2019*, pages 9592–9600, 2019. 3
- [4] P. Bergmann, M. Fauser, D. Sattlegger, and C. Steger. Uninformed students: Student-teacher anomaly detection with discriminative latent embeddings. In *2020 IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR)*, pages 4182–4191, 2020. 2, 3, 7
- [5] J. Božič, D. Tabernik, and D. Skočaj. End-to-end training of a two-stage neural network for defect detection. In *2020 25th International Conference on Pattern Recognition*, pages 5619–5626, 2020. 3, 4
- [6] L.-C. Chen, G. Papandreou, F. Schroff, and H. Adam. Rethinking Atrous Convolution for Semantic Image Segmentation. *CoRR*, abs/1706.05587, 2017. 3
- [7] X. Chen, D. P. Kingma, T. Salimans, Y. Duan, P. Dhariwal, J. Schulman, I. Sutskever, and P. Abbeel. Variational Lossy Autoencoder. In *International Conference on Learning Representations*, pages 1–17, 2017. 3
- [8] I. Croitoru, S.-V. Bogolin, and M. Leordeanu. Unsupervised learning from video to detect foreground objects in single images. In *International Conference on Computer Vision*, pages 4335–4343, 2017. 3
- [9] A. Diba, V. Sharma, A. Pazandeh, H. Pirsiavash, and L. Van Gool. Weakly supervised cascaded convolutional networks. *Proceedings - 30th IEEE Conference on Computer Vision and Pattern Recognition, CVPR 2017*, 2017-Janua:5131–5139, 2017. 3
- [10] X. Dong, C. J. Taylor, and T. F. Cootes. Defect Detection and Classification by Training a Generic Convolutional Neural Network Encoder. *IEEE Transactions on Signal Processing*, 68(1):6055–6069, 2020. 3, 7
- [11] C. Ge, J. Wang, J. Wang, Q. Qi, H. Sun, and J. Liao. Towards automatic visual inspection: A weakly supervised learning method for industrial applicable object detection. *Computers in Industry*, 121, 2020. 3
- [12] I. Goodfellow, J. Pouget-Abadie, and M. Mirza. Generative Adversarial Networks. In *Neural Information Processing Systems*, pages 1–9, 2014. 3
- [13] Y. Huang, C. Qiu, X. Wang, S. Wang, and K. Yuan. A compact convolutional neural network for surface defect inspection. *Sensors*, 20(7), 2020. 2, 3, 7
- [14] S. Kim, W. Kim, Y. Noh, and F. C. Park. Transfer learning for automated optical inspection. In *2017 International Joint Conference on Neural Networks (IJCNN)*, pages 2517–2524, 2017. 2, 7
- [15] D. P. Kingma, D. J. Rezende, S. Mohamed, and M. Welling. Semi-Supervised Learning with Deep Generative Models. In *Neural Information Processing Systems*, pages 1–9, 2014. 3

- [16] Q. Li, A. Arnab, and P. H. Torr. Weakly- and semi-supervised panoptic segmentation. *Lecture Notes in Computer Science*, 11219 LNCS:106–124, 2018. 3
- [17] H. Lin, B. Li, X. Wang, Y. Shu, and S. Niu. Automated defect inspection of LED chip using deep convolutional neural network. *Journal of Intelligent Manufacturing*, pages 1–10, 2018. 1, 2, 3
- [18] Z. Lin, H. Ye, B. Zhan, and X. Huang. An efficient network for surface defect detection. *Applied Sciences*, 10:6085, 09 2020. 2, 3, 7
- [19] G. Liu, N. Yang, L. Guo, S. Guo, and Z. Chen. A one-stage approach for surface anomaly detection with background suppression strategies. *Sensors*, 20:1829, 03 2020. 7
- [20] Y. Liu, Y. Yuan, C. Balta, and J. Liu. A light-weight deep-learning model with multi-scale features for steel surface defect classification. *Materials*, 13(20):1–13, 2020. 3, 7
- [21] J. Masci, U. Meier, D. Ciresan, J. Schmidhuber, and G. Fricout. Steel defect classification with Max-Pooling Convolutional Neural Networks. In *Proceedings of the International Joint Conference on Neural Networks*, pages 1–6, 06 2012. 2
- [22] P. Mlynarski, H. Delingette, A. Criminisi, and N. Ayache. Deep learning with mixed supervision for brain tumor segmentation. *Journal of Medical Imaging*, 6:034002, 2019. 2, 3
- [23] T.-N. Nguyen and J. Meunier. Anomaly detection in video sequence with appearance-motion correspondence. In *2019 IEEE/CVF International Conference on Computer Vision (ICCV)*, pages 1273–1283, 10 2019. 7
- [24] D. M. Onchis and G.-r. Gillich. Stable and explainable deep learning damage prediction for prismatic cantilever steel beam. *Computers in Industry*, 125:103359, 2021. 1, 2
- [25] D. Pathak, P. Krähenbühl, and T. Darrell. Constrained Convolutional Neural Networks for Weakly Supervised Segmentation Multi-class Image Segmentation. In *International Conference on Computer Vision*, pages 1–28, 2015. 3
- [26] D. Pathak, E. Shelhamer, J. Long, and T. Darrell. Fully Convolutional Multi-Class Multiple Instance Learning. In *International Conference on Learning Representations Workshop*, number 1, pages 1–4, 2015. 3
- [27] D. Rački, D. Tomažević, and D. Skočaj. A compact convolutional neural network for textured surface anomaly detection. In *IEEE Winter Conference on Applications of Computer Vision*, pages 1331–1339, 2018. 2, 3, 7
- [28] O. Ronneberger, P. Fischer, and T. Brox. U-Net: Convolutional Networks for Biomedical Image Segmentation. In *Medical Image Computing and Computer-Assisted Intervention – MICCAI 2015*, pages 234–241, 2015. 3
- [29] F. Saleh, M. S. A. Akbarian, M. Salzmann, L. Petersson, S. Gould, and J. M. Alvarez. Built-in foreground/background prior for weakly-supervised semantic segmentation. *Lecture Notes in Computer Science*, 9912 LNCS:413–432, 2016. 3
- [30] T. Schlegl, P. Seeböck, S. M. Waldstein, G. Langs, and U. Schmidt-Erfurth. f-AnoGAN: Fast unsupervised anomaly detection with generative adversarial networks. *Medical Image Analysis*, 54:30–44, 2019. 3
- [31] T. Schlegl, P. Seeböck, S. M. Waldstein, U. Schmidt-Erfurth, and G. Langs. Unsupervised anomaly detection with generative adversarial networks to guide marker discovery. *Lecture Notes in Computer Science*, 10265 LNCS:146–147, 2017. 3
- [32] T. Schlegl, P. Seeböck, S. M. Waldstein, G. Langs, and U. Schmidt-Erfurth. f-anogan: Fast unsupervised anomaly detection with generative adversarial networks. *Medical Image Analysis*, 54:30 – 44, 2019. 7
- [33] N. Souly, C. Spampinato, and M. Shah. Semi Supervised Semantic Segmentation Using Generative Adversarial Network. In *International Conference on Computer Vision*, pages 5688–5696, 2017. 2, 3
- [34] B. Staar, M. Lütjen, and M. Freitag. Anomaly detection with convolutional neural networks for industrial surface inspection. *Procedia CIRP*, 79:484 – 489, 2019. 2, 3, 7
- [35] D. Tabernik, S. Šela, J. Skvarč, and D. Skočaj. Segmentation-based deep-learning approach for surface-defect detection. *Journal of Intelligent Manufacturing*, pages 1–18, 2019. 1, 2, 3, 4, 6, 7
- [36] F. Wan, C. Liu, W. Ke, X. Ji, J. Jiao, and Q. Ye. C-MIL: Continuation Multiple Instance Learning for Weakly Supervised Object Detection. In *Computer Vision and Pattern Recognition*, volume 1, pages 2199–2208, 2019. 3
- [37] F. Wan, P. Wei, J. Jiao, Z. Han, and Q. Ye. Min-Entropy Latent Model for Weakly Supervised Object Detection. *Proceedings of the IEEE Computer Society Conference on Computer Vision and Pattern Recognition*, pages 1297–1306, 2018. 3
- [38] T. Wang, Y. Chen, M. Qiao, and H. Snoussi. A fast and robust convolutional neural network-based defect detection model in product quality control. *The International Journal of Advanced Manufacturing Technology*, 94(9):3465–3471, Feb 2018. 2, 3, 7
- [39] X. Wang, K. He, and A. Gupta. Transitive Invariance for Self-supervised Visual Representation Learning. In *International Conference on Computer Vision*, 2017. 3
- [40] D. Weimer, B. Scholz-Reiter, and M. Shpitalni. Design of deep convolutional neural network architectures for automated feature extraction in industrial inspection. *CIRP Annals - Manufacturing Technology*, 65(1):417–420, 2016. 1, 2, 3, 5, 7
- [41] Y. Yang, R. Yang, L. Pan, J. Ma, Y. Zhu, T. Diao, and L. Zhang. A lightweight deep learning algorithm for inspection of laser welding defects on safety vent of power battery. *Computers in Industry*, 123:103306, 2020. 1, 3
- [42] J. Yu, X. Zheng, and J. Liu. Stacked convolutional sparse denoising auto-encoder for identification of defect patterns in semiconductor wafer map. *Computers in Industry*, 109:121–133, 2019. 1, 2
- [43] V. Zavrtanik, M. Kristan, and D. Skočaj. Reconstruction by inpainting for visual anomaly detection. *Pattern Recognition*, 112:107706, 2021. 2
- [44] J. Zhang, H. Su, W. Zou, X. Gong, Z. Zhang, and F. Shen. Cadn: A weakly supervised learning-based category-aware object detection network for surface defect detection. *Pattern Recognition*, 109:107571, 2021. 2, 3, 7

- [45] R. Zhang, P. Isola, and A. A. Efros. Split-Brain Autoencoders: Unsupervised Learning by Cross-Channel Prediction. In *Computer Vision and Pattern Recognition*, pages 645–654, 07 2017. 3
- [46] B. Zhou, A. Khosla, A. Lapedriza, A. Oliva, and A. Torralba. Learning Deep Features for Discriminative Localization. In *Computer Vision and Pattern Recognition*, pages 2921–2929, 2016. 3
- [47] Y. Zhu, Y. Zhou, H. Xu, Q. Ye, D. Doermann, and J. Jiao. Learning Instance Activation Maps for Weakly Supervised Instance Segmentation. *IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*, pages 3116–3125, 2019. 2, 3