Journal of Intelligent Manufacturing (2023) 34:3091–3108
https://doi.org/10.1007/s10845-022-02000-4

Automatic quality inspection in additive manufacturing using semi-supervised deep learning

Siyamalan Manivannan[1]

Received: 22 January 2022 / Accepted: 20 July 2022 / Published online: 6 August 2022

Abstract
Ensuring quality production is essential in additive manufacturing processes such as selective laser sintering (SLS). Deep learning based automated systems may provide a better way for real time monitoring to ensure quality production. These systems are usually trained in a fully supervised manner which require large amount of labeled data. Obtaining labeled training data in large quantities is a tedious and time consuming process. To overcome this, a novel semi-supervised deep learning approach is proposed in this work, which can be trained using both labeled and unlabeled data, and hence, reducing the manual labeling efforts needed to train the system. Experimental results on a SLS powder bed defect detection dataset show that the proposed approach is the new state-of-the-art, and shows its potential as a standalone real time monitoring system for SLS. In this dataset the proposed approach beats the state-of-the-art accuracy of 98% with only 25% of the labeled training data compared to other approaches. In addition, an extensive set of experiments were conducted on three additional public defect inspection datasets (NEU steel surface defects, KolektorSDD surface images of plastic electronics commutators, and surface textures) to show the applicability of the proposed approach on other computer integrated manufacturing environments for quality inspection. In all of these datasets the proposed approach beats the state-of-the-art results with relatively small amounts of training data compared to other approaches, which shows the effectiveness of the proposed approach for real time, automated, and accurate quality inspection.

Keywords Additive manufacturing · Computer-integrated manufacturing · Semi-supervised learning · Surface defect recognition

Introduction

Selective laser sintering (SLS) is one of the most popular rapid additive manufacturing process which produces complex and accurate prototypes for various industrial applications including aerospace, electronics, and medical. In SLS, a laser is used as the power source to melt and fuse powdered material (e.g., Nylon or Polyamide) in a layer by layer manner based on the computer-aided designed (CAD) model of a part. Here, a thin layer of metal powder is spread over a build plate, and then a laser is used to selectively melt the powder in locations corresponding to a 2D slice of a 3D part (Scime & Beuth, 2018). After the completion of one layer, the build plate is lowered, and another layer of powder is spread over the existing powder bed, and the process repeats until all the layers of the 3D part are fabricated (Scime & Beuth, 2018).

Ensuring part quality is an essential process in SLS. The quality of the parts fabricated by SLS is not only determined by the fusion of the powder particles of successive layers, but also by the integrity of the powder bed and the stability of the powder application (Xiao et al., 2020). Although a uniformly distributed powder bed, without irregularities, is desirable for a good part quality (Westphal & Seitz, 2021), various irregularities known as powder bed defects may present at different stages of the fabrication process, which include foreign bodies (Westphal & Seitz, 2021), powder accumulations (Westphal & Seitz, 2021; Chen et al., 2021), and incomplete powder spreading (Scime & Beuth, 2018; Xiao et al., 2020). These defects may lead to defective parts production, which will significantly waste materials, and increase the production cost. If the defects are detected in its early stages, timely actions could be taken to ensure quality production.

✉ Siyamalan Manivannan
siyam@univ.jfn.ac.lk

1 Department of Computer Science, Faculty of Science, University of Jaffna, Jaffna, Sri Lanka

In Computer-Integrated Manufacturing, various automated vision based monitoring systems have been proposed for different industrial applications, including, additive manufacturing (Westphal & Seitz, 2021; Baumgartl et al., 2020; Gobert et al., 2018; Scime & Beuth, 2018; Xiao et al., 2020), steel surface inspection (He et al., 2019; Tao et al., 2018; Liu et al., 2020; Di et al., 2019; Lv et al., 2020; Soukup & Huber-Mörk, 2014; Zheng et al., 2020; Gao et al., 2020), fabric/texture inspection (Mei et al., 2018; Zheng et al., 2020), tile inspection (Rudolph et al., 2020), Aluminum profile surface inspection (Liu et al., 2021), inspection of electronic commutators (Tabernik et al., 2019; Xu et al., 2020). These monitoring systems aim to provide real time monitoring and alarm the production line if they found any defects, and hence, they provide a way to ensure quality of the production, reduce wastage of the materials, and therefore, increase the profit. However, the majority of the work proposed for this purpose (e.g. Westphal & Seitz, 2021; Baumgartl et al., 2020; Gobert et al., 2018; Scime & Beuth, 2018; Tao et al., 2018; Xiao et al., 2020; Liu et al., 2020; Soukup & Huber-Mörk, 2014; Tabernik et al., 2019) are supervised approaches, which require large amount of annotated data for training the system. Obtaining annotations in large quantities is a tedious, time consuming, expensive process, and requires expert knowledge. Semi-supervised learning approaches, on the other hand, can be used as an alternate for this purpose as they can be trained with little amount of labeled and large amounts of unlabeled data.

In this work, a simple, and efficient semi-supervised Convolutional Neural Network (CNN) based deep learning approach for the real-time inspection of powder bed defects in SLS is proposed. The proposed approach uses a loss function that minimizes both the entropy of the labeled and the weighted pseudo-labeled data, and in addition, applies entropy regularization. The selection of highly-confident, less noisy pseudo-labels plays an important role in determining the success of the pseudo-label based semi-supervised learning approaches. Therefore, this work investigates different approaches for weighting the contribution of unlabeled samples for semi-supervised learning based on their prediction confidence. A *margin* based weighting scheme (Sect. 3) is proposed to determine how well the samples are predicted and weights are applied for each unlabeled sample based on this. Experiments on a public SLS powder bed defect inspection dataset (Westphal & Seitz, 2021) show the effectiveness of the proposed approach compared to the state-of-the-art. In addition, extensive amount of experiments on three other defect detection datasets [(NEU surface defects (Song & Yan, 2013), KolektorSDD surface images of plastic electronics commutators (Tabernik et al., 2019), and Surface Textures (Huang et al., 2020)] were conducted to show the applicability of the proposed approach to other computer integrated manufacturing environments. New state-of-the-art results were obtained on all of these datasets, particularly with small amount of labeled data used for training compared to other approaches. The experiments prove the potential of the proposed approach for a real time accurate monitoring system for quality inspection.

This paper significantly extends the preliminary work of Mayuravaani and Manivannan (2021). It sets the proposed method in the context of the related literature, describes it in more detail, presents extensive experiments on four public defect recognition datasets and summarizes the performance in experimental comparisons with other methods.
I claim the following contributions:

- To the best of my knowledge, this work is the first pseudo-labeling based deep semi-supervised learning approach for powder bed defect inspection in SLS.
- A novel semi-supervised deep learning approach for the classification of surface defects for industrial automation.
- Investigation of different sample weighting schemes to weight the contribution of each unlabeled sample for semi-supervised training.
- A novel way to weight the contribution of each unlabeled sample for semi-supervised training based on how well that sample is predicted for the pseudo-labeled class compared to other classes.
- Significant amount of experiments on four public datasets to validate the proposed system.

Experiments show that the proposed system beats other state-of the-art systems with relatively small amount of labeled data used for training on all the four public datasets compared to other approaches.

In the following, first the related work is summarized in Sect. 2, and then the proposed methodology is explained in detail in Sect. 3. Section 4 reports the datasets, summarizes the experiments and the results with discussion. Section 5 concludes this work.

Related work

The amount of work focusing on automated approaches for defect inspection in additive manufacturing is relatively low compared to the approaches proposed for other manufacturing environments such as steel or fabric manufacturing. In this section, the work related to automated surface inspection in additive and other manufacturing domains and the recent semi-supervised approaches proposed in Computer Vision literature are reviewed.

Defect inspection in additive manufacturing

Various machine learning approaches were explored in additive manufacturing for different problems including quality assessment and defect classification. This section mainly focuses on powder bed defect detection in SLS.

The majority of the approaches proposed for powder bed defect detection are fully-supervised approaches. Both hand-designed features (Gobertetal, 2018; Scime & Beuth, 2018) and deep learning approaches (Westphal & Seitz, 2021; Scime & Beuth, 2018; Zhang et al., 2018; Xiao et al., 2020; He et al., 2016; Kwon et al., 2020) were explored for this purpose. Gobert et. al. 2018 proposed an approach based on different filters and Support Vector Machine (SVM) classifiers for defect inspection during metallic powder bed fusion. Scime and Beuth 2018 used a filter-bank based approach to classify powder bed images into six categories such as recoater, hopping, recoater streaking, debris, super-elevation, part failure, and incomplete spreading. Here, a set of filters were used for feature extraction, and the extracted features were then clustered to build a dictionary using a k-means clustering algorithm. Classification was done based on comparing the test features with the dictionary.

Inspired by the success of deep learning for image classification, recently, different deep learning based fully supervised approaches were explored for powder bed defect inspection. For example, a CNN based classification method is proposed in Westphal and Seitz (2021) to classify powder bed images into either defective or defect-free. Different CNN architectures such as VGG-16 and Xception were explored for this purpose. A system to detect anomalies of melt tracks based on the features extracted from CNN and a SVM classifier was proposed in Zhang et al. (2018). Multi-scale and multi-stage CNN approaches were also explored: e.g., A multi-scale CNN is proposed in Scime and Beuth 2018 which uses image patches of multiple scales as the input to CNN for powder bed anomaly classification. In Xiao et al. (2020) a two stage CNN model is proposed to detect defects such as warpage, part shifting, and short feed. Here, a Residual neural net (ResNet He et al., 2016) was used to extract features at different scales, and then this multi-scale features were processed by a region proposal net to generate potential defective regions, and finally a fully connected layer is used to get the final prediction.

Only a few semi-supervised deep learning approaches were explored compared to the fully supervised approaches for powder bed defect classification. In Okaro et al. (2019) a Gaussian Mixture Model (GMM) based approach was proposed, where the GMM was first trained using both the labeled and unlabeled data, and then a Bayesian classifier was used for classification. The Π model from the Temporal Ensemble method (Laine & Aila, 2016) was used in Yuan et al. (2019) as a semi-supervised approach for Monitoring SLS process.

Defect inspection in other manufacturing domains

Various image processing and shallow learning based approaches were initially explored for defect detection in various manufacturing domain such as aluminum (Win et al., 2015), fabric (Mallik-Goswami & Datta, 2000; Shumin et al, 2011), electronics (Bai et al., 2014), steel (Song & Yan, 2013), etc. E.g., a contrast adjustment thresholding method was proposed for defect detection in aluminum surfaces in Win et al. (2015). A morphological based image processing approach was proposed for fabric defect detection in Mallik-Goswami and Datta (2000). In shallow learning, first a set of hand designed features were extracted from the images and then a machine learning classifier was used to predict whether the given image contains defect or not. Various hand designed features such as Local Binary Patterns (Song & Yan, 2013), Histogram of Oriented Gradients (Shumin et al., 2011) and classifiers such as Nearest Neighbor Song and Yan (2013), Support Vector Machines Shumin et al. (2011) and Song and Yan (2013) were explored for this purpose. These traditional approaches usually contain many parameters which require expert knowledge to determine, and do not work well in practice compared to the modern deep learning based approaches.

Deep learning based approaches, particularly CNNs, have the ability to learn complex decision boundaries, therefore, perform significantly better than the traditional image processing and shallow learning based approaches, and report state-of-the-art results in various domains including computer aided manufacturing. However, they often rely on large quantities of training data. To overcome this, transfer learning based approaches were explored, which takes a pretrained CNN model (e.g., trained on a large scale image dataset such as ImageNet Tan et al. (2018)) and fine tune it to suit for the target application. CNN based approaches have already been explored for different manufacturing applications (Xu et al., 2020; Liu et al., 2020; Soukup & Huber-Mörk, 2014; Cha et al., 2018; Lv et al., 2020; Tabernik et al., 2019; Ren et al., 2018), including, defect detection in motor commutators (Xu et al., 2020), defect detection in steel surfaces (Soukup & Huber-Mörk, 2014), weld defect detection (Ren et al., 2018), wood defect detection (Ren et al., 2018), etc. These approaches either propose new or use off-the-shelf CNN architectures. However, most of these approaches are supervised approaches, requiring labeled training data. As discussed in Sect. 1 obtaining large amount of labeled data for training is a difficult and time consuming task. To overcome this, semi-supervised learning approaches were also explored for automatic surface inspection, which include different learning strategies (Wang et al., 2021; Gao et al., 2020; Hajizadeh et al., 2016; Guo et al., 2020), and CNN architectures (Zheng et al., 2020; Rudolph et al., 2020).

Semi-supervised approaches proposed in computer vision

Although Semi-supervised deep learning approaches were not well explored for defect inspection in automated manufacturing, they are heavily explored in computer vision and machine learning literature for various problems such as image recognition (Berthelot et al., 2019; Sohn et al., 2020), Natural Language Processing (Liang, 2005), etc. *Consistency regularization* and *Pseudo-labeling* are two prominent categories of semi-supervised learning approaches (Rizve et al., 2021) among the others (Van Engelen & Hoos, 2020).

Consistency regularization techniques enforce the output predictions of an input and it's augmented/perturbed versions to be similar to each other. Several methods were used to produce perturbed inputs, Dropout (Srivastava et al., 2014) and random data augmentations (Sohn et al., 2020; Sajjadi et al., 2016) are named a few. *MixMatch* (Berthelot et al., 2019) uses the *Mixup* (Zhang et al., 2017) data augmentation technique for consistency regularization. Temporal ensembling (Laine & Aila, 2016) forces the output predictions of an input to be consistent over epochs. However, consistency regularization techniques heavily rely on domain-specific data augmentations, which are not easy to generate for all data modalities (Rizve et al., 2021).

On the other hand, pseudo-labeling is a technique which augments the labeled training data by automatically labeling the unlabeled data and add them with the original training data. Here, first a model is trained using only the labeled data, and then it was used to predict the labels (pseudo-labels) of the unlabeled data. Next a new model is trained or the existing one is updated using the combination of the original labeled and the pseudo-labeled data. Here, all the pseudo-labeled data or the pseudo-labels with high-confident predictions were used. Usually the quality of the pseudo-labels are more important than their quantity (Sohn et al., 2020). Therefore, different approaches were proposed and they differ from each other in the way the pseudo-labels are generated and selected for training. For example, in Lee (2013) the pseudo-labels were calculated using a single network trained using the labeled data, and then the network was updated by considering all the pseudo-labeled data regardless of their confidence. Since the pseudo-labels are often noisy, training the network using all the pseudo-labels will lead to noisy training (Rizve et al., 2021). Hence, only the pseudo-labels with high confidence predictions (confidence above a threshold) were widely considered for better training (Rizve et al., 2021; Sohn et al., 2020). Instead of these hard-thresholding, various soft-thresholding based approaches were also proposed, which apply weights for each input based on its prediction confidence (Mo et al., 2021; Shi et al., 2018; Ren et al., 2020); E.g., in (Mo et al., 2021) an Information Entropy based technique was used to determine the weights; In Shi et al. (2018) the density of the local neighborhood around each sample was considered when determining the weight as outliers may lie in the low density areas compared to informative samples. Recently, an uncertainty aware pseudo-labeling selection method was proposed in Rizve et al. (2021) and reported improved performance over many other pseudo-labeling techniques including MixMatch (Berthelot et al. 2019). Here the uncertainty estimation of a particular image was computed based on a set of predictions obtained from the network for that image.

Compared to the above approaches, this work proposes a pseudo-labeling based semi-supervised deep learning approach with a margin-based criteria to weight the unlabeled samples for training the deep learning model and shows state-of-the-art performance on different defect inspection datasets.

Proposed methodology

Assume that the training set comprised of a set of labeled ($\mathcal{D}_L$) and a set of unlabeled ($\mathcal{D}_U$) images. Let $\mathbf{x}_i$ be an image and $\mathbf{y}_i = \{0, 1\}^C$ be the one-hot representation of its label, if $\mathbf{x}_i$ is from the labeled data, and its label is unknown if it is from the unlabeled data, where C represents the total number of classes. Lets assume that $\mathbf{p}_i = [p_{i1}, p_{i2}, \ldots, p_{iC}]$ be the probability of the image $\mathbf{x}_i$ belonging to different classes, and it is obtained from the CNN classifier by applying a softmax function on the corresponding outputs, i.e., $p_{ij} = \frac{exp(z_{ij})}{\sum_k \exp(z_{ik})}$, where, $\mathbf{z}_i = [z_{i1}, \ldots, z_{iC}]$ is the output of the network for $\mathbf{x}_i$.

The predicted label (*pseudo label*), $\hat{\mathbf{y}}_i$, of $\mathbf{x}_i$ is obtained as the class which gives the maximum probability, and therefore, the c^{th} element of $\hat{\mathbf{y}}_i$ can be given as:

$$\hat{y}_{ic} = \begin{cases} 1, & \text{if } c = \operatorname{argmax}_{c'}\{p_{ic'}\} \\ 0, & \text{otherwise} \end{cases} \tag{1}$$

In addition, assume that $\hat{p}_i$ be the maximum probability of $\mathbf{x}_i$ over all the classes.

$$\hat{p}_i = \max\,(p_{i1}, p_{i2}, \ldots, p_{iC}) \tag{2}$$

Loss function

The parameters of the CNN are usually learned in a fully supervised manner using the following *Cross Entropy* loss function.

$$\mathcal{L}_L = -\frac{1}{|\mathcal{D}_L|}\sum_{c=1}^{C} u_c \sum_{i\in\mathcal{D}_L} y_{ic}\log p_{ic} \tag{3}$$

where, $\mathcal{D}_L = \{\mathbf{x}_i, \mathbf{y}_i\}$ be the labeled training set, u_c is the class weight associated with the class c to handle imbalanced number of training samples from different classes, and it could be set as the inverse of the image frequencies of each class.

However, as the training data comprised of both labeled and unlabeled images, the following loss function is minimized.

$$\mathcal{L} = \mathcal{L}_L + \mathcal{L}_U + \alpha\mathcal{R} \tag{4}$$

$$\text{where, } \mathcal{L}_U = -\frac{1}{|\mathcal{D}_U|}\sum_{c=1}^{C} u_c \sum_{i\in\mathcal{D}_U} w_i\ \hat{y}_{ic}\log p_{ic} \tag{5}$$

$$\text{and } R = -\frac{1}{|\mathcal{D}_L| + |\mathcal{D}_U|}\sum_{i=1}^{N}\sum_{c=1}^{C} p_{ic}\log p_{ic} \tag{6}$$

This loss function contains three terms. The first term ($\mathcal{L}_L$) minimizes the cross entropy of the labeled images (Eq. 3). The second term ($\mathcal{L}_U$) is based on the unlabeled data, which minimizes the cross entropy of the pseudo-labeled images. Here, the pseudo-labels of the unlabeled images were considered as if they were true labels, and the probabilities of the images belonging to these pseudo-classes are maximized. Based on these probability values, the weight (w_i) of each unlabeled image is calculated, and this weight determines how much an image contributes to this loss. The last term ($\mathcal{R}$) is an entropy regularization term proposed in Tanaka et al. (2018) to concentrate the probability of each image on a single class. This term penalizes the images which have flat probability distributions (similar probabilities for belonging to different classes), and therefore, encourages each image to focus on a single class to improve the probability of that image belonging to that class. In this way the images will be pushed away from the decision boundary between different classes to improve the network training. In Eq. 4 α is a parameter which controls the contribution of the entropy regularization term on the overall loss (refer Sect. 4.4.2 for the effect of this parameter on the classification results).

Weighting schemes

In Eq. 5, w_i is the weight which indicates the confidence on the pseudo-label of an unlabeled image $\mathbf{x}_i$. The following sections explain different ways to determine these weights.

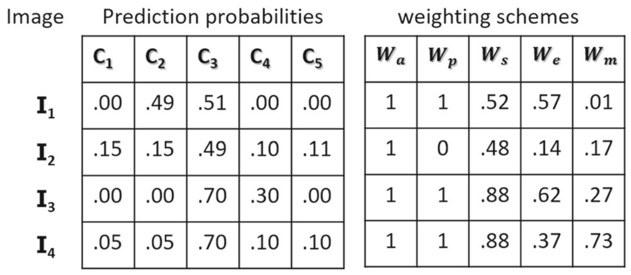


Image	Prediction probabilities					weighting schemes				
	C_1	C_2	C_3	C_4	C_5	W_a	W_p	W_s	W_e	W_m
I_1	.00	.49	.51	.00	.00	1	1	.52	.57	.01
I_2	.15	.15	.49	.10	.11	1	0	.48	.14	.17
I_3	.00	.00	.70	.30	.00	1	1	.88	.62	.27
I_4	.05	.05	.70	.10	.10	1	1	.88	.37	.73

Fig. 1 Weight for different unlabeled images ($I_1, \ldots, I_4$) determined by different weighting schemes, where $C_1, \ldots, C_5$ represent different classes.

Equal weights for all the unlabeled images (W_a)

A recent work (Gao et al., 2020) for defect classification assumes that the pseudo-labels obtained for all the unlabeled images are correct, and sets $w_i = 1, \forall i$. This approach considers all the unlabeled images and their corresponding pseudo-labels regardless of their prediction confidence. However, sometimes the pseudo-labels of the unlabeled images could be noisy. For example, the pseudo-label of the image I_1 in Fig. 1 could be a wrong one as it has very similar probabilities for belonging to the classes C_2 and C_3. If the pseudo-labels of many unlabeled images are noisy, the network may learn a wrong decision boundary leading to performance drop.

Selection of a subset of unlabeled images based on $\hat{p}_i$ (W_p)

To avoid having wrong pseudo-labels for training, only the unlabeled images with larger $\hat{p}_i$ values can be considered. i.e., the weights are determined by applying a threshold t on $\hat{p}_i$, i.e., $w_i = 1$ if $\hat{p}_i > t$, and 0 otherwise. Here, some unlabeled images may not contribute to learning as they have low prediction probabilities than the threshold. In addition, the selected set of samples based on this scheme is sensitive to the threshold value. For example, although I_1 and I_2 of Fig. 1 have similar prediction probabilities, only the image I_1 will be selected due to hard thresholding (when $t = 0.5$).

Soft-weighting based on $\hat{p}_i$ (W_s)

In this case the weights are defined using the following function to avoid the problems associated with hard thresholding:

$$w_i = \frac{1}{1 + e^{-\beta(\hat{p}_i - t)}} \tag{7}$$

where, β and t control the softness of the weights. In this scheme the prediction confidences belonging to different classes were not considered. E.g., the images I_3 and I_4 of Fig. 1 have the same weights by this scheme, although they have different prediction confidences. I define the prediction confident as how well an image is predicted as one class compared to the others.

The proposed soft-weighting scheme which uses a margin criteria (W_m)

Determining the weights just based on $\hat{p}_i$ (Sects. 3.2.2 and 3.2.3) may not be a good idea. Instead, the weights must be determined based on how well an image is predicted as one class compared to the other classes.

Lets consider the images I_1 and I_2 (Fig. 1). As they are predicted with similar probability ($\hat{p}$) values they receive similar weights by the weighting scheme W_s. However, image I_2 should be considered as a high confident prediction than image I_1 as the probability of image I_2 belonging to other classes (other than its pseudo-label class) are very small. Similarly, I_4 should receive a higher weight than I_3, as I_4 is predicted with higher confident. Therefore, I define the weight as:

$$w_i = \frac{1}{1 + e^{-\beta(d-t)}} \qquad \text{where,} \quad d = \hat{p}_i - \hat{p}'_i \tag{8}$$

where, $\hat{p}'_i$ is the second maximum probability for the image $\mathbf{x}_i$, and the softness of the weights are defined by the parameters β and t.

Entropy based weighting(W_e)

According to Information theory, information entropy is a measure of uncertainty. Entropy has a value close to zero for high confident predictions, and its value increases with prediction uncertainty. Therefore, entropy can be used to determine sample weights. An information entropy based sample weighting scheme was recently proposed in Mo et al. (2021) for semi-supervised learning. The sample weights are determined as follows:

$$w_i = 1 - \frac{E(\mathbf{p}_i)}{\log(C)} \tag{9}$$

where, $\mathbf{p}_i = [p_{i1}, \ldots, p_{iC}]$ is the probability of image $\mathbf{x}_i$, and C represents the number of classes. The entropy of $\mathbf{p}_i$, i.e., $E(\mathbf{p}_i)$, can be defined as:

$$E(\mathbf{p}_i) = -\sum_{i=1}^{C} p_{ic} \log(p_{ic}) \tag{10}$$

where p_{ic} is the probability of image $\mathbf{x}_i$ belonging to class c.

However, this approach does not provide reasonable weights. For example, a high weight value is given by this approach to image I_1 (Fig. 1) compared to image I_2 (0.57 vs 0.14), even though image I_2 has a high confident prediction, and image I_1 has a very noisy prediction having approximately equal probabilities to two different classes. In this scenario, image I_1 must receive a weight close to zero, and this weight must be very smaller compared to the weight of image I_2. However, this entropy based weighting assigns a very high weight to image I_1, and a lower weight to image I_2 which are very much inappropriate. Similarly, a high weight is assigned for image I_3 than image I_4, although the weights should be assigned in the other way. In addition, although the image I_4 is predicted with a higher probability value compared to image I_1, this scheme provides a high weight value to image I_1 than image I_4, proving its inefficiency in determining sample weights.

The direct use of these entropy-based weights (Eqn 9) may lead to worst classification performance, particularly, for the datasets which have large number of classes. This is mainly because when the predictions become sparse, the entropy term becomes smaller. In addition, the denominator term becomes large for the datasets which have large number of classes. In this scenario, all the weights, regardless of prediction score become arbitrarily high. This leads to the selection of large number of noisy labels, and hence, worst classification performance. To avoid this, in the reported experiments a threshold function was applied on these weights to get binary weights (0 or 1), and then these binary weights were used when optimizing the semi-supervised loss.

Summary of the weighting schemes

From Fig. 1 we can see that W_a (Sect. 3.2.1) considers all the samples regardless of their prediction confidence, and assign equal weights. In the experiments it is found that this scheme leads to worse classification performance, and therefore the results based on this scheme are not reported in Sect. 4. W_p (Sect. 3.2.2) neither make use of all the images, nor consider prediction confidence of different classes when determining weights. W_s (Sect. 3.2.3) assigns similar weights to the images I_1 and I_2 (and also I_3 and I_4), although each of them is predicted with different prediction confidence. W_e assigns high weights to image I_1 although its prediction is noisy. In addition, it assigns high weight to image I_3 compared to image I_4, although image I_4 should get high weight as it is predicted with high confidence. Compared to all the other approaches, the proposed approach (W_m in Sect. 3.2.4) weights the images based on their prediction confidence and gives higher weights to the images which are predicted with high confident, and low weights on the other hand.

The training procedure of the proposed approach

The proposed self-learning approach initially uses a fully supervised training to learn the parameters of CNN, i.e., only

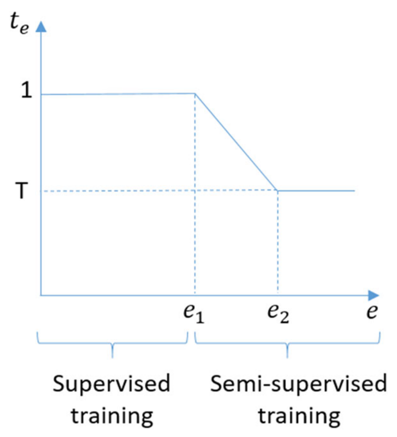


Fig. 2 Threshold over epochs

the labeled training data is used to train the CNN with the loss function defined in Eq. 3 for the first e_1 epochs (e.g., $e_1 = 75$). After that, the prediction probability of each of the unlabeled sample is calculated, and then each unlabeled sample is weighted using the proposed approach given by Eq. 8. These weighted unlabeled samples are then added with the labeled samples to update the network with the proposed loss function defined in Eqn 4. At each epoch e, the weights are recalculated again. Instead of applying a fixed threshold t in Eq. 8, here, an alternative approach is used to determine the value of t to enable a smooth training. In this approach, a high threshold (i.e, $t = 1$) was set immediately after the fully supervised training (i.e., $e = e_1$), and then this value was linearly reduced until it reaches the specified value T (e.g., $T = 0.9$), at the end of e_2 epochs. In the experiments, e_2 is set as the first milestone (refer Table 2), where the learning rate was reduced by a factor of 0.1. The threshold at the epoch e, t_e, can be given as,

$$t_e = \begin{cases} \frac{(T-1)(e-e_1)}{e_2-e_1} + 1, & e_1 < e < e_2 \\ T, & e \geq e_2 \end{cases} \tag{11}$$

This procedure is illustrated in Fig. 2. Here, the threshold is set to 1 for supervised training as no unlabeled data is included, and then the threshold is linearly reduced to the target value to progressively include high confident pseudo-labeled data, which enables a smooth training procedure.

Experiments and results

Dataset, experimental settings, results and discussion are summarized in this section.

Datasets and experimental setting

Four public datasets were used to show the effectiveness of the proposed approach. Please refer Table 1 for more details about these datasets. Figures 3, 4, 5 and 6 show sample images from different categories of these datasets.

SLS powder bed dataset

This dataset contains images of powder bed surface from an SLS printing system, and was introduced and well studied in Westphal and Seitz (2021). This dataset contains two categories of images: defective and defect-free. Sample images from each of these categories are shown in Fig. 3. The images were preprocessed in Westphal and Seitz (2021) by removing the unwanted background information and resized to 180×180. The training and the test sets respectively contain 1000 and 500 images per category.

Northeastern University dataset (NEU)

The NEU dataset (Song & Yan, 2013) contains six types of steel surface defects: crazing, inclusion, patches, pitted surface, rolled-in scale, and scratches. Each category of this dataset contains 300 grayscale images of size 200×200 pixels. Following (He et al., 2019) 60% of the images from each category were considered for training and the rest were used for testing.

KolektorSDD dataset

This dataset (Tabernik et al., 2019) contains a total of 399 surface images of plastic electronics commutators, among them 52 images show clear visible defects and the rest (347 images) are defect-free. Each image is in gray scale, and has a size of approximately 1408×512 pixels. The images were resized to have a size of 352×128 pixels, and $\frac{2}{3}$ of the images were used for training and rest were used for testing.

Surface textures dataset

This dataset (Huang et al., 2020) contains a total of 8, 674 images from 64 classes, which is collected in Huang et al. (2020) from three public datasets; KTH surface dataset Caputo et al. (2005) (3, 194 images from 11 classes), Kyberge dataset (Kylberg, 2011) (4, 480 images from 28 classes), and UIUC dataset (Lazebnik et al., 2005)) (1, 000 images from 25 classes). The classes include wood, blanket, cloth, leather, so forth, and they are very commonly seen in surface inspection problems (Huang et al., 2020). The number of images in each class varies from 40 to 513. Each image in this dataset has a size of 331×331 pixels. In this work, all the images were converted to grayscale as only a few portion of the orig-

Table 1 Details of different datasets used in this research (DF - Defect–free, D - Defective)

Dataset	Evaluation measure	Total no of images	train:test split / class	No of images for testing	No of labeled training images under different setting				
					5%	10%	25%	50%	100%
SLS (2 classes)	Accuracy	3000 (1500 img/class)	2 : 1	1000 (500 img/class)	100 (50 img/class)	200 (100 img/class)	500 (250 img/class)	1000 (500 img/class)	2000 (1000 img/class)
NEU (6 classes)	Accuracy	1800 (300 img/class)	6 : 4	720 (120 img/class)	36 (9 img/class)	108 (18 img/class)	270 (45 img/class)	540 (90 img/class)	1080 (180 img/class)
KolektorSDD (2 classes)	Average Precision	399 (347 DF, 52 D)	2 : 1	132 (115 DF, 17 D)	14 (12 DF, 2 D)	28 (24 DF, 4 D)	67 (58 DF, 9 D)	134 (116 DF, 18 D)	267 (232 DF, 35 D)
Textures (64 classes)	Accuracy	8675 (40 to 513 img/class)	1 : 1	4335 (20 to 256 img/class)	222 (1 to 13 img/class)	440 (2 to 26 img/class)	1090 (5 to 65 img/class)	2173 (10 to 129 img/class)	4340 (20 to 257 img/class)

inal images are in color. Following Huang et al. (2020) half of the images were considered for training and the rest were used for testing.

For all the above datasets only the image level labels were considered, and no color information was used.

Data augmentation and dropout

To avoid overfitting and to improve the generalization ability of the CNN, an extensive amount of data augmentation and dropout were used at the training stage, and neither data augmentation nor dropout were used at the testing stage. Data augmentation includes, random horizontal, vertical flipping, random rotations ($\pm 180°$), color jitter (brightness, contrast, saturation and hue were changed) and random Affine transformations (scaling and translation). However, for the KolektorSDD dataset, all the above augmentations were used except $\pm 180°$ rotations, instead, rotations of $\pm 20°$ were applied as the shape of the images are not square. For the SLS dataset as following Westphal and Seitz (2021) random horizontal flipping, random rotations ($\pm 20°$), random shifting by a factor of ± 0.2 and random scaling by a factor of ± 0.15 were used. In addition, for all the datasets a dropout layer (with a probability of 0.5) was used immediately before the fully connected layer of the network. Images were normalized (zero mean and unit variance) before feeding them to the CNN.

Network architecture, training and evaluation measures

An ImageNet pretrained ResNet (He et al. 2016) architecture with 10 layers and 18 layers were used as the baseline CNN architectures for the smaller (NEU and KolektorSDD) and the larger (Surface Textures) datasets respectively. Although, the proposed approach can be applicable to any CNN architecture, ResNet was chosen as it is widely used. Stochastic Gradient Descent was used to optimize the network weights. Table 2 lists the settings used for training the CNN; E.g., for the NEU dataset, the learning rate was initially set to 0.01 for the first 150 epochs, and then it was reduced by a factor of 0.1 from 150 to 200 epochs and again it was reduced by the same factor after 200 epochs, while the total number of epochs was set to 250. In the case of semi-supervised training, the network is trained for the first 75 epochs in a fully supervised manner, and then unlabeled images were added as explained in Sect. 3.3.

Accuracy for the NEU and Surface Texture datasets, and *Average Precision* for the KolektorSDD dataset were used as the evaluations measures as they were widely on these datasets (Di et al., 2019; He et al., 2019; Gao et al., 2020; Huang et al., 2020; Xu et al., 2020) to compare different methods. Each experiment was repeated three times (unless

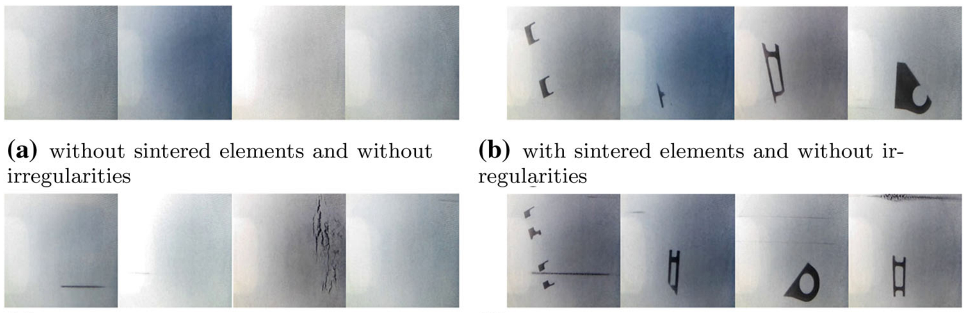


Fig. 3 Sample powder bed images from the SLS dataset

(a) without sintered elements and without irregularities

(b) with sintered elements and without irregularities

(c) without sintered elements and with irregularities

(d) with both sintered elements and irregularities

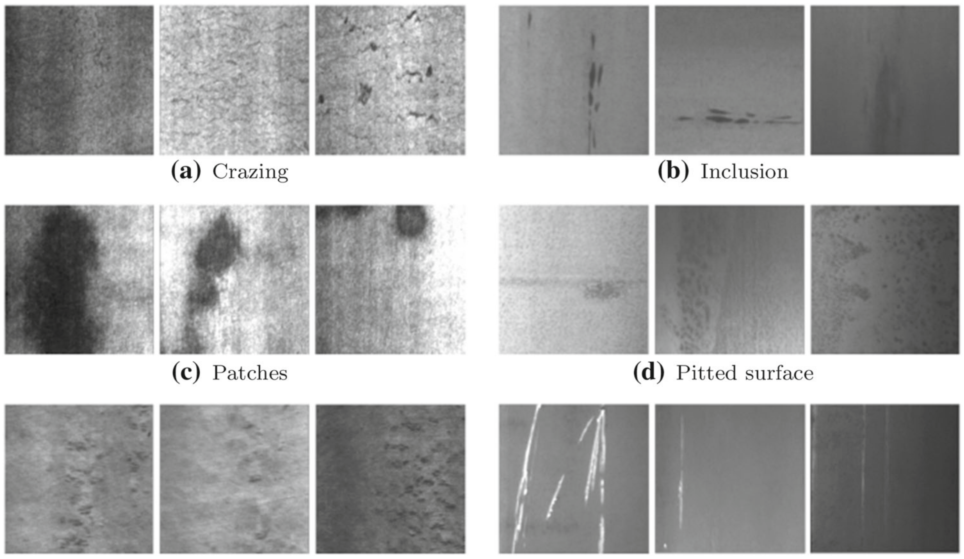


Fig. 4 Sample images from the NEU dataset

(a) Crazing

(b) Inclusion

(c) Patches

(d) Pitted surface

(e) Rolled-in scale

(f) Scratches

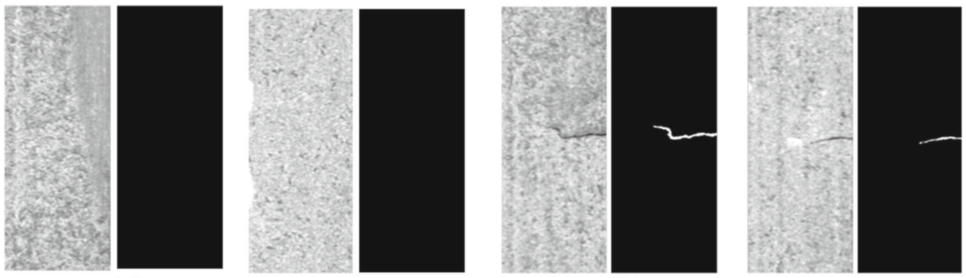


Fig. 5 Example images and their corresponding segmentation masks from the KolektorSDD dataset. Note that, the segmentation masks are shown here only to illustrate the defective regions, and are not used in any of the experiments

(a) Defect–free images

(b) Defective images

otherwise specified), and the mean and the standard deviation of the evaluation measures over these iterations were reported. In all the semi-supervised learning based experiments, t and β (Eq. 8) were set to 0.9 and 30 respectively. The regularization parameter (α, Eq. 4) was set to 1, 1, 0.1, and 10 respectively for the SLS, NEU, KolektorSDD, and the Surface Texture datasets. These parameters were selected based on some initial experiments conducted on a part of the training set. Please refer Sect. 4.4.2 for the effect of these parameters on the classification results.

In all of the following experiments, s% of training images were randomly selected as the labeled training set, and the rest were used as the unlabeled dataset, where s is varied from 5% to 100% (Table 1). For the unlabeled images the original labels were discarded, and only the images were considered.

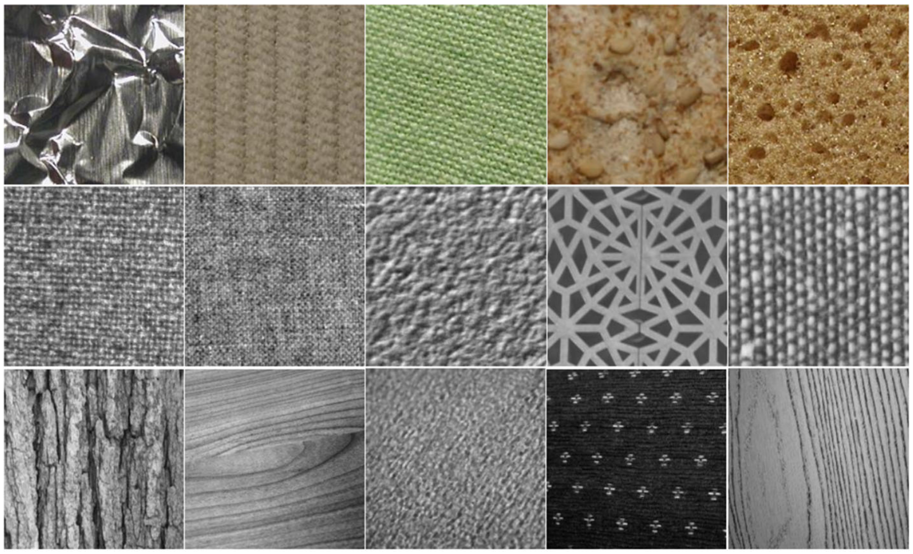


Fig. 6 Example images from different categories of the Surface Texture dataset

Table 2 Network training details (SS - semi-supervised learning)

Dataset	batch size	initial learning rate	no. of supervised epochs for SS	milestones	total no. of epochs
SLS	64	0.01	25	50,80	100
NEU	64	0.01	75	150,200	250
KolektorSDD	64	0.05	75	125,170	180
Surface Textures	64	0.005	75	150,200	225

Table 3 Performance comparison of fully supervised (FS) vs. semi-supervised (SS) approaches on different datasets

Dataset	Backbone	FS/SS	Overall accuracy (%) for different number of training samples per class				
			5%	10%	25%	50%	100%
SLS	Resnet-10	FS	76.40 ± 6.02	92.45 ± 0.95	97.50 ± 0.34	97.50 ± 0.26	**98.01 ± 0.02**
		SS	**82.63 ± 0.59**	**96.91 ± 0.52**	**97.57 ± 0.03**	**98.00 ± 0.03**	–
NEU	Resnet-10	FS	91.69 ± 1.52	97.00 ± 0.09	99.45 ± 0.26	99.67 ± 0.17	**99.86 ± 0.11**
		SS	**98.34 ± 0.36**	**99.50 ± 0.12**	**99.75 ± 0.14**	**99.82 ± 0.05**	–
KolektorSDD	Resnet-10	FS	79.94 ± 4.37	85.59 ± 1.13	**100 ± 0.00**	**100 ± 0.00**	**100 ± 0.00**
		SS	**85.10 ± 2.24**	**88.60 ± 0.69**	**100 ± 0.00**	**100 ± 0.00**	–
Textures	Resnet-18	FS	86.71 ± 0.45	93.58 ± 0.38	96.62 ± 0.32	99.09 ± 0.05	**99.52 ± 0.13**
		SS	**89.52 ± 0.48**	**95.33 ± 0.28**	**98.60 ± 0.20**	**99.50 ± 0.01**	–

The best scores are marked with bold

Results and discussion

Table 3 compares the fully supervised baseline with the proposed semi-supervised approach on all the three datasets. The proposed semi-supervised approach gives significant boost in performance compared to its fully supervised counterpart on all of these datasets. For example, $\sim 6\%$, $\sim 7\%$, 6% and 3% performance improvements were obtained respectively on the SLS, NEU, KolektorSDD and Surface Texture datasets, when 5% of the labeled training data is used for training. On the NEU dataset, the proposed semi-supervised approach obtains an accuracy of 98.34 ± 0.36 only with 5% of labeled data, which is comparable with the accuracy (99.45 ± 0.26) obtained by the full supervised approach with 25% of labeled data. On this dataset, the proposed semi-supervised approach achieves an accuracy of 99.50 ± 0.12 only with 10% of labeled training data. Similarly, on the Surface Textures dataset, the semi-supervised approach obtains an accuracy of 95.33 ± 0.28 with only 10% of the labeled training data, which is comparable with the accuracy (96.62 ± 0.32) obtained by the fully supervised approach with 25% labeled training data.

Table 4 The performance of fully supervised (FS) and the proposed semi-supervised (SS) approaches with and without pretraining (PT) and/or data augmentation (Aug)

PT	Aug	FS/SS	Overall accuracy (%) for different number of training samples per class				
			9 (5%)	18 (10%)	45 (25%)	90 (50%)	180 (100%)
✗	✗	FS	80.88 ± 0.85	86.84 ± 0.16	94.86 ± 0.24	97.72 ± 0.09	99.45 ± 0.06
		SS	79.70 ± 0.80	89.88 ± 0.45	96.27 ± 0.12	98.46 ± 0.70	–
✗	✓	FS	88.90 ± 0.01	91.48 ± 0.52	94.86 ± 0.22	98.90 ± 0.09	99.62 ± 0.03
		SS	89.33 ± 0.62	93.28 ± 0.17	96.52 ± 0.10	99.14 ± 0.04	–
✓	✗	FS	91.20 ± 0.96	95.83 ± 0.11	98.62 ± 0.28	99.32 ± 0.02	99.75 ± 0.06
		SS	94.28 ± 2.20	98.96 ± 0.03	99.13 ± 0.15	99.54 ± 0.26	–
✓	✓	FS	91.69 ± 1.52	97.00 ± 0.09	99.45 ± 0.26	99.67 ± 0.17	$\mathbf{99.86 \pm 0.11}$
		SS	$\mathbf{98.54 \pm 0.36}$	$\mathbf{99.50 \pm 0.12}$	$\mathbf{99.75 \pm 0.14}$	$\mathbf{99.82 \pm 0.05}$	–

The best scores are marked with bold

Table 5 Effects of different parameters on the NEU dataset when 5% of the labeled training data is used with the proposed semi-supervised learning approach

(a)Effect of t and α when β is fixed to $\beta = 30$

α / t	0	0.1	1	10
0.95	94.93 ± 0.51	96.11 ± 0.71	97.32 ± 1.1	84.61 ± 7.69
0.9	96.71 ± 0.22	95.87 ± 1.10	98.34 ± 0.36	92.18 ± 1.98
0.8	97.31 ± 0.42	97.45 ± 0.57	98.48 ± 0.20	87.56 ± 6.85
0.5	98.05 ± 0.32	97.92 ± 0.66	98.45 ± 0.46	82.71 ± 6.32

(b)Effect of β when $\alpha = 1$ and $t = 0.9$

β	Accuracy
0.1	97.65 ± 0.55
1.0	97.93 ± 0.51
10.0	98.46 ± 0.53
30.0	98.34 ± 0.36
50.0	97.94 ± 0.78

Effect of data augmentation and network pretraining

The effect of parameters

This section investigates the effect of parameters involved in the proposed approach. The results are reported in Table 5, which investigate the effect of t, β (Eq. 8) and α (Eq. 4) on the classification accuracy of the NEU dataset. From Table 5a we can see the importance of the regularization term used in Eq. 4. Adding this term to the loss function improves the classification accuracy when t is fixed. However, larger value of α (i.e., $\alpha = 10$) degrades the classification performance on this dataset as more weight is given for regularization than classification (Eq. 4). When t is considered, there is a trade-off between the number of samples which are selected from the unlabeled data in the pseudo-labeling process and their correctness. A smaller t value (e.g., $t = 0.5$) allows the CNN to select a larger number of pseudo-labeled data, on the other hand, a larger t (e.g., $t = 0.95$) value allow to select relatively small number, but highly accurate labeled data. I found that $t = 0.9$ is a good trade-off parameter when all the three datasets are considered, although, on the NEU dataset (Table 5a) smaller t values lead to better classification results. Table 5b reports the accuracy values when changing β, while fixing the other parameters constant. $\beta = 10$ and 30 give improved results than the others.

Comparison of the weighting schemes

Figure 8 compares different sample weighting schemes that are discussed in Sect. 3.2 on the Surface Textures dataset. Here, the network was trained in a fully supervised manner for 75 epochs with 5% of labeled training data, and then the trained CNN was used to predict the rest of the images in the training dataset. The sample weights for these testing images based on different sample weighting approaches (W_m, W_s and W_e) were calculated, and different threshold values were applied to these weights to calculate the accuracy of the images which pass this thresholding criteria. The results are reported in Fig. 8, which shows that the proposed approach selects more correctly labeled images than

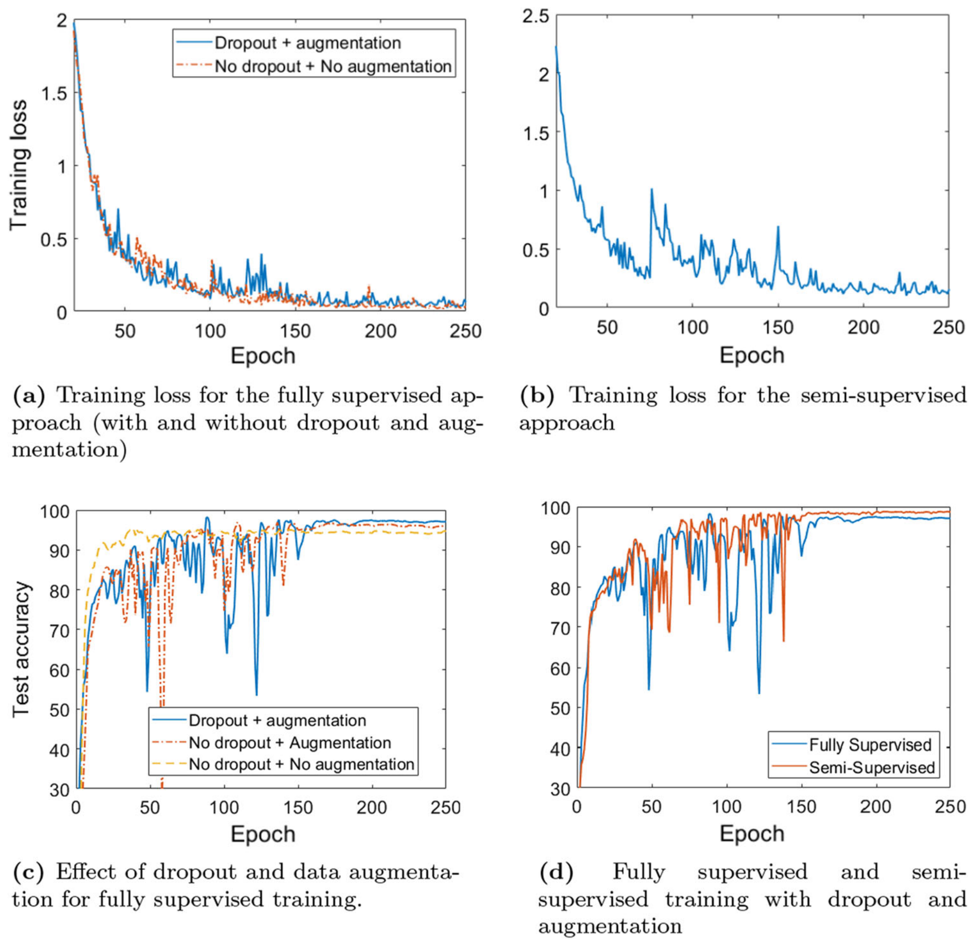


Fig. 7 Training loss and the testing accuracy over different training epochs under different setting of the NEU dataset with 10% of labeled training data. (Best viewed in color.)

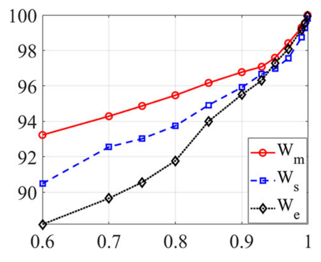


Fig. 8 Comparison of different weighting schemes: X-axis represents threshold values, and Y-axis represents accuracy

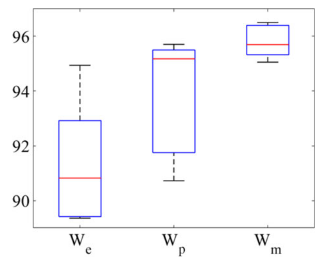
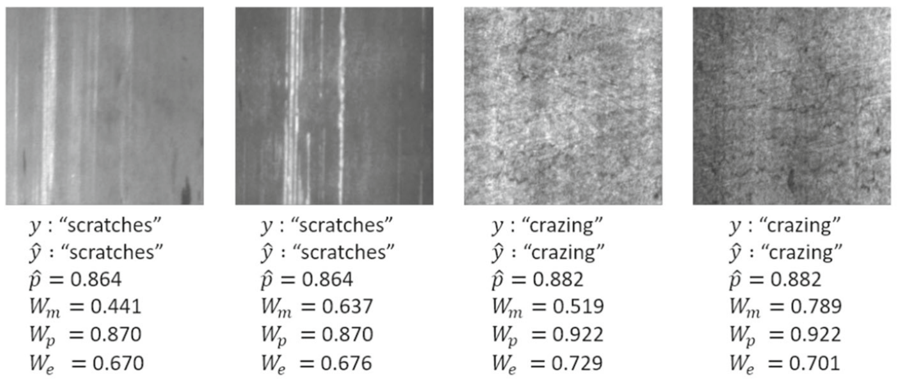


Fig. 9 Box plot comparing different weighting schemes

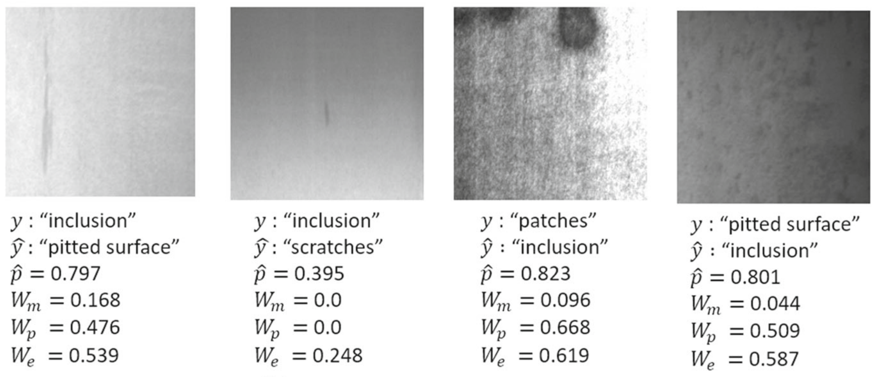
the others, proving its effectiveness in selecting more correctly labeled samples from the unlabeled training images. The prediction accuracy of all the predicted images (without applying a threshold, i.e., $t = 0$) by this trained CNN (W_a, in Sect. 3.2.1) was also evaluated, and found that only $\sim 60\%$ of images were correctly predicted. This shows the ineffectiveness of the weighting scheme W_a. Note that, this weighing scheme is used by a recent work, PLCNN (Gao et al., 2020), for surface defect recognition, and in Table 8 I show that the proposed approach outperforms PLCNN with a significant margin on the NEU dataset.

In Fig. 8 when the threshold is set to a very high value (e.g. 0.99) all the methods performs equally well, but note that, when the threshold is very high only few samples will be selected, which will slow down the semi-supervised training. On the other hand, a small threshold value (e.g., 0.5) will select a very large number of less accurate samples, which will lead to noisy training. Therefore, the selection of the threshold is a trade-off between the amount of pseudo-labeled samples selected and their correctness.

Figure 9 reports the *Box plot* to compare different weighting schemes on the Surface Texture dataset, when 5% of labeled training data are considered. This Box plot sum-



(a) Some correctly predicted images.



(b) Some wrongly predicted images.

Fig. 10 Weights assigned by different weighing schemes for the unlabeled images of the NEU dataset. Here, y, $\hat{y}$ and $\hat{p}$ represents the true label, pseudo-label, and the probability of the particular image belonging to $\hat{y}$ respectively

maries the results of five repeated experiments. We can observe that the proposed approach W_m performs better than other approaches, suggesting its suitability for the selection of more accurate samples from the unlabeled data for semi-supervised training.

Figure 10 shows some of the correctly classified and wrongly classified images with their corresponding weights assigned by different approaches. Although the predicted probabilities are same (first two and the last two images of Fig.10a) the proposed weighting scheme assigned different weights by considering the top two probabilities of each image. However, W_p assigns the same weights for these images as it is only based on the top probability. On the other hand, W_e assigns large weights to the wrongly classified images compared to W_m and W_s.

Comparison with the state-of-the-art approaches

In this section, the proposed approach is compared with the $\Pi -$ model (Laine & Aila, 2016) - a widely-used consistency regularization based semi-supervised approach, and the Uncertainty-Aware Pseudo-Labling Selection method (Rizve et al., 2021)—a recently proposed pseudo-labeling based semi-supervised approach which reports better performance than many other semi-supervised approaches in Rizve et al. (2021). In addition, the proposed approach is also compared with the approaches proposed in the computer aided manufacturing literature to show that the proposed approach is the new state-of-the-art for defect detection in computer aided manufacturing.

Π $-$ model (Laine and Aila 2016) is one of the well-known approach, which makes the outputs of different augmented versions of inputs to be similar to each other, and uses the following loss function for optimization.

$$\mathcal{L} = \mathcal{L}_L + \delta(t)\ \mathcal{L}_c \tag{12}$$

where, $\mathcal{L}_L$ is the supervised cross-entropy loss defined in Eq. 3, and $\mathcal{L}_c$ is the consistency regularization term, defined as:

$$\mathcal{L}_c = \frac{1}{N}\sum_{i=1}^{N} \|\mathbf{p}_i - \mathbf{p}_i'\|^2 \tag{13}$$

where, $\mathbf{p}_i$ and $\mathbf{p}_i'$ are the output probabilities obtained for two different augmentations of the same image. In Eq. 12 $\delta(t)$ is a trade-off parameter, determined as $\delta(t) = \exp\left[-5(1-T)^2\right]$ Laine and Aila (2016), and in the experiments it was set to $T = \frac{t}{100}$.

Uncertainty-Aware Pseudo-Labling Section (UPS) (Rizve et al. 2021) is a recent method which selects less noisy pseudo-labels based on an uncertainty estimation. In addition, this method also uses negative labels for training. The loss function used by UPS for a pseudo-labeled image is given as:

$$\mathcal{L}_i = -\frac{1}{s_i}\sum_{c=1}^{C} g_{ic}\left[\hat{y}_{ic}\log(p_{ic}) + (1-\hat{y}_{ic})\log(1-p_{ic})\right] \tag{14}$$

where, $g_{ic} \in \{0, 1\}$ determines whether the pseudo-label of the i^{th} image belonging to class c (i.e., $\hat{y}_{ic}$) is selected or not, and $s_i = \sum_c g_{ic}$ is the number of selected pseudo-labels for sample i. g_{ic} is defined as:

$$\begin{aligned} g_{ic} = {} & \mathbb{1}\left[u(p_{ic}) \le \kappa_p\right]\mathbb{1}\left[p_{ic} \ge \tau_p\right] \\ & + \mathbb{1}\left[u(p_{ic}) \le \kappa_n\right]\mathbb{1}\left[p_{ic} \le \tau_n\right] \end{aligned} \tag{15}$$

Here, $u(p_{ic})$ is the uncertainty associated with p_{ic}. A positive pseudo-label is selected if its probability score is high ($p_{ic} \ge \tau_p$) and certain ($u(p_{ic}) \le \kappa_p$). Conversely, a negative pseudo-label is considered, if the network is sufficiently confident of a class's absence (i.e., $p_{ic} \le \tau_n$ and $u(p_{ic}) \le \kappa_n$). In Rizve et al. (2021) dropout was used during the inference time to estimate the uncertainty of the prediction of a particular sample. The uncertainty was obtained as the standard deviation of the network's predictions for 10 forward passes of that sample. The values for κ_p, τ_p, κ_n and τ_n were set to 0.05, 0.9, 0.005 and 0.05 respectively. To have a fair comparison, first the model was trained in a fully supervised manner (Eq. 3), and then updated in a semi-supervised manner by using loss functions defined in Eqs. 14 and 3.

Table 6 reports the comparative results on the NEU dataset, while all the other experimental settings were fixed (e.g. CNN backbone is fixed to ResNet-10). Both the Π − model and UPS improve the classification performance over the fully supervised baseline. However, the proposed approach performs significantly better than both of them, proving its efficiency (both in selecting high-confident pseudo-labels and the loss function). Note that, UPS requires several forward passes of the same image through the network to determine the uncertainty value, and similarly, the Π − model requires to get the predictions of augmented versions of the input images, both of which are time consuming. On the other hand, the proposed approach is simple and yet gives better performance.

In this section the proposed approach is compared with the approaches proposed for the defect detection in computer aided manufacturing. Tables 7, 8, 9 and 10 compare the performance of the proposed approach compared to the state-of-the-art approaches on the SLS powder bed, NEU, KolektorSDD, and the Surface Textures datasets respectively. On all of these datasets, the proposed approach achieves the state-of-the-art results with relatively low amount of labeled data for training. For example, on the NEU dataset, the state-of-the-art result was obtained with only 10% of labeled training data. Note that, on the same dataset, the Generative Adversarial Network (GAN) based approach cDCGAN (He et al., 2019) generates additional images, and train the system using all the labeled (100%) and the generated images. How-

Table 6 Comparison with Π − model and UPS on the NEU dataset with different percentage of labeled training data

Method	5%	10%
FS	91.69 ± 1.52	97.00 ± 0.09
Π − model (Laine and Aila, 2016)	95.22 ± 0.71	97.74 ± 0.25
UPS (Rizve et al., 2021)	96.61 ± 0.71	98.57 ± 0.23
Proposed	**98.34 ± 0.36**	**99.50 ± 0.12**

The best scores are marked with bold

Table 7 Comparison with the state-of-the-art approaches on the SLS dataset

Method	% labeled	Acc	Precision	Recall	F1 score	ROC-AUC
Proposed (FS)	5%	76.40 ± 6.02	.851 ± .134	.665 ± .007	.742 ± .048	.854 ± .052
Proposed (SS)	5%	82.63 ± 0.59	.930 ± .030	.708 ± .014	.803 ± .002	.870 ± .017
Proposed (FS)	10%	92.45 ± 0.95	.992 ± .001	.851 ± .003	.916 ± .001	.983 ± .007
Proposed (SS)	10%	96.91 ± 0.02	.986 ± .001	.929 ± .001	.956 ± .001	.996 ± .001
Proposed (FS)	25%	97.50 ± 0.26	.963 ± .001	.988 ± .006	.975 ± .003	.996 ± .002
Proposed (SS)	**25%**	**98.00 ± 0.03**	**.965 ± .001**	**.982 ± .001**	**.976 ± .001**	**.996 ± .002**
Proposed (FS)	100%	**98.01 ± 0.03**	**.963 ± .001**	**.990 ± .002**	**.975 ± .001**	**.996 ± .001**
VGG-16 (Westphal & Seitz, 2021) (FS)	100%	97.1	.963	.980	.972	.993
Xception (Westphal & Seitz, 2021) (FS)	100%	89.4	.876	.918	.897	.934

The best scores are marked with bold

Table 8 Comparison with the state-of-the-art approaches on the NEU dataset

Methods	Overall accuracy (%) for different number of training samples per class				
	9 (5%)	18 (10%)	45 (25%)	90 (50%)	180 (100%)
Supervised learning					
Zhou et. al. (2017)	–	–	78.09	80.00	86.64
Li et. al. (2016)	–	–	82.81	85.39	95.00
Ren et. al. (2018)	–	–	–	90.88	92.04
He et. al. (CAE-SGAN) Di et al. (2019)	–	–	–	–	98.96
He al. el. (cDCGAN) He et al. (2019)	–	–	–	–	99.56
Wang et. el. (MMGCN) Wang et al. (2021)	–	–	–	–	99.72
Proposed	**91.69 ± 1.52**	**97.00 ± 0.09**	**99.45 ± 0.26**	**99.67 ± 0.17**	**99.86 ± 0.11**
Semi-supervised learning					
He et. al. (CAE-SGAN) Di et al. (2019)	–	–	85.83	94.87	–
He el. al. (cDCGAN) He et al. (2019)	–	–	89.58	96.06	-
Gao et. al. (PLCNN) Gao et al. (2020)	–	–	90.7	–	–
Wang et. al. (MMGCN) Wang et al. (2021)	–	–	98.06	98.75	–
Proposed	**98.54 ± 0.36**	**99.50 ± 0.12**	**99.75 ± 0.14**	**99.82 ± 0.05**	–

The best scores are marked with bold

Table 9 Comparison with the state-of-the-art approaches on KolektorSDD dataset

Method	AP for different number of positive training samples		
	5	10	33
Segmentation based approaches which use both image and pixel level labels for training			
Jakob et. al. (2020)	96.71	99.31	100
Tabernik et. al. (2019)	95.80	98.80	99.00
Cognex ViDi (commercial software) Tabernik et al. (2019)	89.20	95.60	99.00
Image level labels only			
Xu et. al. (2020)	–	98.0	99.50
Proposed	88.60 ± 0.69	**100 ± 0.00**	**100 ± 0.00**

The best scores are marked with bold

Table 10 Comparison with the state-of-the-art approaches on Surface Textures

Method	Accuracy for different percentage of labeled training images		
	25%	50%	100%
Huang et. al. (2020)	–	–	99.33
Proposed	**98.60 ± 0.20**	**99.50 ± 0.01**	**99.52 ± 0.13**

The best scores are marked with bold

ever, it is interesting to see that the proposed approach, which uses only 5% of labeled training data beats this GAN based approach with a significant margin, proves its efficiency. On the KolektorSDD dataset, proposed approach achieves the state-of-the-art results with 25% of labeled training data. Note that, the approaches (Božič et al., 2020) and (Tabernik et al., 2019) (Table 9) use both the image and the pixel-level labels for training the system. But the approach proposed, on the other hand, uses image-level labels only. In addition, on the Surface Textures dataset, the state-of-the-art results were achieved with only 50% of the labeled training data.

Discussion

The proposed approach is efficient in terms of the required amount of human annotations, computational time, memory requirements, and classification performance. For example, on the SLS powder bed dataset the approach proposed in Westphal and Seitz (2021) reports an accuracy of 97.1% when trained with 2000 labeled training images and with VGG-16 model which contains approximately 138M parameters to train. On the other hand, the proposed approach beats the approach proposed in Westphal and Seitz (2021)

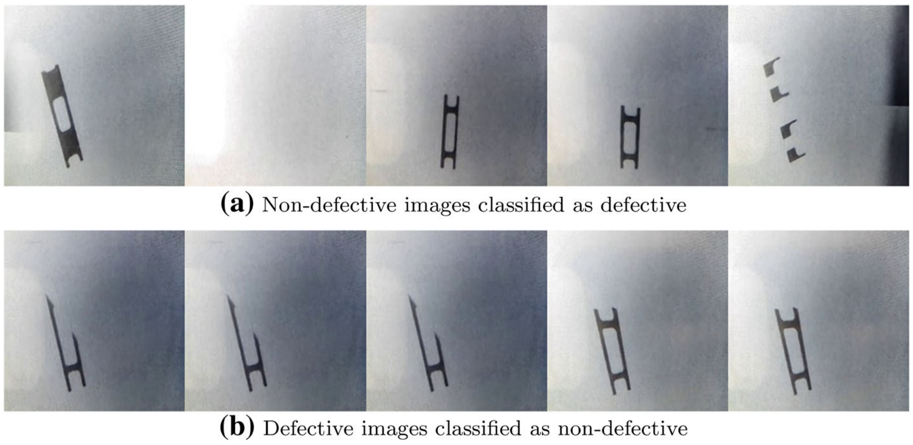


Fig. 11 Sample images from the SLS powder bed dataset which are misclassified by the proposed approach

(a) Non-defective images classified as defective

(b) Defective images classified as non-defective

with only 500 labeled images (and 1500 unlabeled images) for training and with a much simpler ResNet-10 model which contains only $\sim$ 5.17M parameters, thereby reducing the computational and memory requirements. The proposed approach requires only 0.9 seconds to process 1000 images on a NVidia P100 GPU with a memory of 16GB enabling a real time defect monitoring system. Some of the miss-classified images from the SLS powder bed dataset by the proposed approach are shown in Fig. 11. These miss-classifications are reasonable, as in this figure, the non-defective images contain some patterns which make them look like defective, and in the defective images there is no visible defective patterns.

Conclusion

In this work, a novel semi-supervised deep learning approach is proposed for the detection of powder bed defects in additive manufacturing. The selection of highly-confident pseudo-labels plays an important role in determining the success of pseudo-labeling based semi-supervised learning approaches. Therefore, a margin-based weighing scheme is proposed to weight the contribution of the unlabeled samples based on their prediction confidence and showed that it performs better than other weighting schemes. Extensive experiments with four public datasets (SLS powder bed, NEU steel surface defects, KolektorSDD surface images of plastic electronics commutators and Surface Textures) show the effectiveness of the proposed approach, and its applicability to other computer integrated manufacturing environments. The proposed approach achieves the new state-of-the-art results on all of these datasets with relatively little amount of labeled training data compared to other approaches and enables an automated, accurate, real time surface defect monitoring.

Declarations

Conflict of interest The author declares that he has no conflict of interest.

References

Bai, X., Fang, Y., Lin, W., Wang, L., & Ju, B. (2014). Saliency-based defect detection in industrial images by using phase spectrum. *IEEE Transactions on Industrial Informatics, 10*(4), 2135–2145.

Baumgartl, H., Tomas, J., Buettner, R., & Merkel, M. (2020). A deep learning-based model for defect detection in laser-powder bed fusion using in-situ thermographic monitoring. *Progress in Additive Manufacturing, 5,* 277–285.

Berthelot, D., Carlini, N., Goodfellow, I., Papernot, N., Oliver, A., & Raffel, C. A. (2019). MixMatch: A holistic approach to semi-supervised learning. *Advances in Neural Information Processing Systems, 32,* 5049–5059.

Božič, J., Tabernik, D., & Skočaj, D. (2020). End-to-end training of a two-stage neural network for defect detection. arXiv:2007.07676.

Caputo, B., Hayman, E., & Mallikarjuna, P. (2005). Class-specific material categorisation. In *IEEE international conference on computer vision* (Vol. 2, pp. 1597–1604).

Cha, Y.-J., Choi, W., Suh, G., Mahmoudkhani, S., & Büyüköztürk, O. (2018). Autonomous structural visual inspection using region-based deep learning for detecting multiple damage types. *Computer Aided Civil and Infrastructure Engineering, 33,* 731–747.

Chen, Y., Peng, X., Kong, L., Dong, G., Remani, A., & Leach, R. (2021). Defect inspection technologies for additive manufacturing. *International Journal of Extreme Manufacturing, 3*(2), 022002.

Di, H., Ke, X., Peng, Z., & Dongdong, Z. (2019). Surface defect classification of steels with a new semi-supervised learning method. *Optics and Lasers in Engineering, 117,* 40–48.

Gao, Y., Gao, L., Li, X., & Yan, X. (2020). A semi-supervised convolutional neural network-based method for steel surface defect recognition. *Robotics and Computer-Integrated Manufacturing, 61,* 101825.

Gobert, C., Reutzel, E. W., Petrich, J., Nassar, A. R., & Phoha, S. (2018). Application of supervised machine learning for defect detection during metallic powder bed fusion additive manufacturing using high resolution imaging. *Additive Manufacturing, 21,* 517–528.

Guo, J., Wang, Q., & Li, Y. (2020). Semi-supervised learning based on convolutional neural network and uncertainty filter for façade defects classification. *Computer-Aided Civil and Infrastructure Engineering,* 1–17.

Hajizadeh, S., Núñez, A., & Tax, D. M. (2016). Semi-supervised rail defect detection from imbalanced image data. In *14th IFAC symposium on control in transportation systems* (Vol. 49, No. 3, pp. 78–83).

He, K., Zhang, X., Ren, S., & Sun, J. (2016) Deep residual learning for image recognition. In *IEEE conference on computer vision and pattern recognition* (pp. 770–778).

He, Y., Song, K., Dong, H., & Yan, Y. (2019). Semi-supervised defect classification of steel surface based on multi-training and generative adversarial network. *Optics and Lasers in Engineering, 122,* 294–302.

Huang, Y., Qiu, C., Wang, X., Wang, S., & Yuan, K. (2020). A compact convolutional neural network for surface defect inspection. *Sensors, 20*(7), 1974.

Kwon, O., Kim, H. G., Ham, M. J., Kim, W., Kim, G.-H., Cho, J.-H., et al. (2020). A deep neural network for classification of melt-pool images in metal additive manufacturing. *Journal of Intelligent Manufacturing, 31*(2), 375–386.

Kylberg, G. (2011). The Kylberg texture dataset v. 1.0," Tech. Rep. 35, Centre for Image Analysis, Swedish University of Agricultural Sciences and Uppsala University, Uppsala, Sweden.

Laine, S, & Aila, T. (2016). Temporal ensembling for semi-supervised learning. arXiv:1610.02242.

Lazebnik, S., Schmid, C., & Ponce, J. (2005). A sparse texture representation using local affine regions. *IEEE Transactions on Pattern Analysis and Machine Intelligence, 27*(8), 1265–1278.

Lee, D.-H., et al. (2013).Pseudo-label: The simple and efficient semi-supervised learning method for deep neural networks. In *Workshop on challenges in representation learning, ICML* (Vol. 3, p. 896).

Liang, P. (2005). *Semi-supervised learning for natural language*. PhD thesis, Massachusetts Institute of Technology.

Liu, J., Song, K., Feng, M., Yan, Y., Tu, Z., & Zhu, L. (2021). Semi-supervised anomaly detection with dual prototypes autoencoder for industrial surface inspection. *Optics and Lasers in Engineering, 136,* 106324.

Liu, Y., Yuan, Y., Balta, C., & Liu, J. (2020). A light-weight deep-learning model with multi-scale features for steel surface defect classification. *Materials,13*(20).

Lv, X., Duan, F., Jiang, J., Fu, X., & Gan, L. (2020). Deep metallic surface defect detection: The new benchmark and detection network. *Sensors, 20*(6), 1562.

Mallik-Goswami, B., & Datta, A. (2000). Detecting defects in fabric with laser-based morphological image processing. *Textile Research Journal, 70,* 758–762.

Mayuravaani, M., & Manivannan, S. (2021) A semi-supervised deep learning approach for the classification of steel surface defects. In *International Conference on Information and Automation for Sustainability*, pp. 179–184.

Mei, S., Yang, H., & Yin, Z. (2018). An unsupervised-learning-based approach for automated defect inspection on textured surfaces. *IEEE Transactions on Instrumentation and Measurement, 67*(6), 1266–1277.

Mo, J., Gan, Y., & Yuan, H. (2021). Weighted pseudo labeled data and mutual learning for semi-supervised classification. *IEEE Access, 9,* 36522–36534.

Okaro, I. A., Jayasinghe, S., Sutcliffe, C., Black, K., Paoletti, P., & Green, P. L. (2019). Automatic fault detection for laser powder-bed fusion using semi-supervised machine learning. *Additive Manufacturing, 27,* 42–53.

Ren, R., Hung, T., & Tan, K. C. (2018). A generic deep-learning-based approach for automated surface inspection. *IEEE Transactions on Cybernetics, 48*(3), 929–940.

Ren, Z., Yeh, R., & Schwing, A. (2020). Not all unlabeled data are equal: Learning to weight data in semi-supervised learning. *Advances in Neural Information Processing Systems, 33,* 21786–21797.

Rizve, M. N., Duarte, K., Rawat, Y. S., & Shah, M. (2021) In defense of pseudo-labeling: An uncertainty-aware pseudo-label selection framework for semi-supervised learning. arXiv:2101.06329.

Rudolph, M., Wandt, B., & Rosenhahn, B. (2020). Same same but differnet: Semi-supervised defect detection with normalizing flows. arXiv:2008.12577.

Sajjadi, M., Javanmardi, M., & Tasdizen, T. (2016). Regularization with stochastic transformations and perturbations for deep semi-supervised learning. *Advances in Neural Information Processing Systems, 29,* 1171–1179.

Scime, L., & Beuth, J. (2018). Anomaly detection and classification in a laser powder bed additive manufacturing process using a trained computer vision algorithm. *Additive Manufacturing, 19,* 114–126.

Scime, L., & Beuth, J. (2018). A multi-scale convolutional neural network for autonomous anomaly detection and classification in a laser powder bed fusion additive manufacturing process. *Additive Manufacturing, 24,* 273–286.

Shi, W., Gong, Y., Ding, C., Tao, Z. M., & Zheng, N. (2018). Transductive semi-supervised deep learning using min-max features. In *Proceedings of the European conference on computer vision* (pp. 299–315).

Shumin, D., Zhoufeng, L., & Chunlei, L. (2011). Adaboost learning for fabric defect detection based on HOG and SVM. In *International conference on multimedia technology* (pp. 2903–2906).

Sohn, K., Berthelot, D., Li, C.-L., Zhang, Z., Carlini, N., Cubuk, E. D., Kurakin, A., Zhang, H., & Raffel, C. (2020). Fixmatch: Simplifying semi-supervised learning with consistency and confidence. arXiv:2001.07685.

Song, K., & Yan, Y. (2013). A noise robust method based on completed local binary patterns for hot-rolled steel strip surface defects. *Applied Surface Science, 285,* 858–864.

Soukup, D., & Huber-Mörk, R. (2014). Convolutional neural networks for steel surface defect detection from photometric stereo images. *International Symposium on Visual Computing,* 668–677.

Srivastava, N., Hinton, G., Krizhevsky, A., Sutskever, I., & Salakhutdinov, R. (2014). Dropout: A simple way to prevent neural networks from overfitting. *The Journal of Machine Learning Research, 15*(1), 1929–1958.

Tabernik, D., Šela, S., Skvarč, J., & Skočaj, D. (2019) Segmentation-based deep-learning approach for surface-defect detection *Journal of Intelligent Manufacturing,31*.

Tan, C., Sun, F., Kong, T., Zhang, W., Yang, C., & Liu, C. (2018) A survey on deep transfer learning. arXiv:1808.01974.

Tanaka, D., Ikami, D., Yamasaki, T., & Aizawa, K. (2018). Joint optimization framework for learning with noisy labels. In *IEEE conference on computer vision and pattern recognition* (pp. 5552–5560).

Tao, X., Zhang, D., Ma, W., Liu, X., & Xu, D. (2018). Automatic metallic surface defect detection and recognition with convolutional neural networks. *Applied Sciences, 8,* 1575.

Van Engelen, J. E., & Hoos, H. H. (2020). A survey on semi-supervised learning. *Machine Learning, 109*(2), 373–440.

Wang, Y., Gao, L., Gao, Y., & Li, X. (2021). A new graph-based semi-supervised method for surface defect classification. *Robotics and Computer-Integrated Manufacturing, 68,* 102083.

Westphal, E., & Seitz, H. (2021). A machine learning method for defect detection and visualization in selective laser sintering based on convolutional neural networks. *Additive Manufacturing, 41,* 101965.

Win, M., Bushroa, A. R., Hassan, M. A., Hilman, N. M., & Ide-Ektessabi, A. (2015). A contrast adjustment thresholding method for surface defect detection based on mesoscopy. *IEEE Transactions on Industrial Informatics, 11*(3), 642–649.

Xiao, L., Lu, M., & Huang, H. (2020). Detection of powder bed defects in selective laser sintering using convolutional neural network. *The International Journal of Advanced Manufacturing Technology, 107,* 2485–2496.

Xu, L., Lv, S., Deng, Y., & Li, X. (2020). A weakly supervised surface defect detection based on convolutional neural network. *IEEE Access, 8,* 42285–42296.

Yi, L., Li, G., & Jiang, M. (2016). An end-to-end steel strip surface defects recognition system based on convolutional neural networks. *Steel Research International, 88*(2), 1600068.

Yuan, B., Giera, B., Guss, G., Matthews, I., & Mcmains, S. (2019) Semi-supervised convolutional neural networks for in-situ video monitoring of selective laser melting. In *IEEE winter conference on applications of computer vision*, pp. 744–753.

Zhang, H., Cisse, M., Dauphin, Y. N., & Lopez-Paz, D. (2017). mixup: Beyond empirical risk minimization. arXiv:1710.09412

Zhang, Y., Hong, G. S., Ye, D., Zhu, K., & Fuh, J. Y. (2018). Extraction and evaluation of melt pool, plume and spatter information for powder-bed fusion am process monitoring. *Materials & Design, 156,* 458–469.

Zheng, X., Wang, H., Chen, J., Kong, Y., & Zheng, S. (2020). A generic semi-supervised deep learning-based approach for automated surface inspection. *IEEE Access, 8,* 114088–114099.

Zhou, S., Chen, Y., Zhang, D., Xie, J., & Zhou, Y. (2017). Classification of surface defects on steel sheet using convolutional neural networks. *Materiali in tehnologije, 51,* 123–131.