Industrial defect detection on the edge with deep learning over scarcely labeled and extremely imbalanced data

Joe Lorentz*, Thomas Hartmann†, Assaad Moawad‡ and Djamila Aouada§
DataThings S.A.
Luxembourg, Luxembourg
*†‡first.last@datathings.com
SnT, University of Luxembourg
Luxembourg, Luxembourg
*§first.last@uni.lu

***Abstract*—Reliable automated defect detection is an integral part of modern manufacturing and improved performance can provide a competitive advantage. Despite the proven capabilities of convolutional neural networks (CNNs) for image classification, application on real world tasks remains challenging due to the high demand for labeled and well balanced data of the common supervised learning scheme. Semi-supervised learning (SSL) promises to achieve comparable accuracy while only requiring a small fraction of the training samples to be labeled. However, SSL methods struggle with data imbalance and existing benchmarks do not reflect the challenges of real world applications. In this work we present a CNN-based defect detection unit for thermal sensors. We describe how to collect data from a running process and release our dataset of 1k labeled and 293k unlabeled samples. Furthermore, we investigate the use of SSL under this challenging real world task. *Index Terms*—quality assurance, edge computation, imbalanced data, semi-supervised learning**

I. Introduction

Defect detection is an important part of modern manufacturing and automating the process with robust computer vision applications provides a competitive advantage [1]. Convolutional neural networks (CNNs) are well established in research, providing state-of-the-art performance on many image classification challenges [2, 3]. Therefore, CNNs are now getting adopted on various quality assurance tasks, such as for solar panels [4] or LED chips [5]. In this work we present a CNN-based defect detection solution for the challenging quality assurance of thermal sensors that requires multiple images to detect defects from different viewing angles. The new machine learning solution should replace the existing quality assurance system on a running industrial production line. The current solution can only differentiate good from defective products and the robustness is insufficient. The new system differentiates 6 types of defects, allowing the manufacturer to identify potential machine faults faster and react accordingly.

This is a pre-print version of the article that is currently undergoing peer review

The present project is supported by the National Research Fund, Luxembourg; Project Reference 14297122

Collecting image datasets specific to the manufacturing domain is very challenging as interference of the production needs to be minimal. Furthermore, a good production is supposed to mainly produce good pieces, leading to problems to identify and collect samples of the minority defect classes. We placed a prototype on a running production line to collect samples and to investigate the feasibility of our CNN-based solution on the edge. Labeling has been a major challenge and our Thermal Sensor Multi-View Defects (TSMVD) dataset, consists of 1k labeled and 293k unlabeled samples.

Recently, the so-called semi-supervised-learning (SSL) approach has been proposed to train CNNs on labeled and unlabeled data simultaneously [6]. State of the art SSL approaches promise comparable performance to fully supervised counterparts while only using a small fraction of the labels [7]. Research solutions for imbalanced SSL [8, 9] only consider imbalance ratios of up to 200 though. Furthermore, class distribution mismatches between labeled and unlabeled data are only considered to a limited extent. In contrast, the class distribution on the unlabeled part of TSMVD is unknown but expected to differ drastically from the labeled distribution. We conduct an early investigation on using SSL for this real world scenario and discuss challenges and potential. To enable researchers to reproduce our results and allow for future work on SSL under extreme imbalance and label scarcity, we make our code and the TSMVD dataset publicly available[1].

The remainder of this paper is organized as follows. Section II presents related work. Section III describes our prototype and the TSMVD dataset is presented in Section IV. Experiments and results are described in Section V followed by a conclusion in Section VI.

II. Related Work

CNNs provide state-of-the-art performance on many image classification benchmarks [2, 3]. Using multiple input images for a single classification decision has been used on various

[1]https://tcdata2206.s3.eu-west-3.amazonaws.com/index.html

tasks, such as 3D shape recognition [10], or x-ray based disease classification [11]. We build upon [10] to pool features from multiple views while only training a single backbone network to keep the computational effort and memory consumption feasible for our embedded process.

Defect detection models are usually evaluated on surface defect datasets, such as NEU [12], PCB [13] or the artificially generated DAGM [14]. Real world defect datasets tend to be small, e.g. PCB and NEU feature less then 2k samples and do not consider unlabeled samples. In contrast, TSMVD offers 1k labeled and 293k unlabeled samples with 3 views each.

A major concern when applying CNNs for real time defect detection is their high computational cost. Due to latency or security concerns, it is usually preferred to handle the time-critical process on the edge rather than outsourcing to a cloud solution [15]. Hardware manufactures have started to design units specifically for running neural networks on the edge. Most notably NVIDIA's Jetson family [16, 17] provides GPU-acceleration for embedded machine learning. We use the low-end Jetson Nano to run the CNN inference in our work.

Class imbalance is a common and well studied challenge of supervised machine learning that leads to classifiers overfitting on the majority class [18]. [19] suggests to directly adjust the logits based on the label distribution to improve performance on imbalanced data. While originally developed for supervised learning, we also use logit adjustment during the supervised part of SSL in our experiments.

Another challenge of supervised machine learning is the high effort for labeling the training data. Transfer learning can be used to pre-train the classifier on data with more-readily available labels [20]. Another important tool to help models to generalize beyond the scope of the given training samples is data augmentation [21]. We use the RandAugment algorithm [22] which offers automated random data augmentation and is used in other related work [7, 8]. More recently, semi-supervised learning [6, 7, 23] (SSL) has been proposed to generate pseudo-labels from more easily available unlabeled data and use them to regularize the models. Approaches to better regularize SSL on imbalanced data exist [8, 9] but are usually evaluated on imbalance ratios of 200 or less and distribution mismatches between labeled and unlabeled data are only considered to a limited extent. [8] only considers cases where the unlabeled imbalance is lower than on the labeled data or in reversed order. In our work, the imbalance ratio on the labeled data is 12.5 while we estimate the ratio on the unlabeled data to be at least 400, and the possibility of it reaching the magnitude of thousands. We argue that in general it is more realistic to assume the labeled data to be more balanced rather then the opposite.

III. Multi-view defect detection on the edge

This work has been done in collaboration with a large supplier for the automotive industry. The targeted manufacturing line produces thermal sensors and features multiple quality assurance systems after sequential production steps. In one step, the negative-temperature-coefficient thermistor (NTC) is attached to a support body via two soldering points. The soldering seams are checked directly after this step and rejected parts are ejected from the production line.

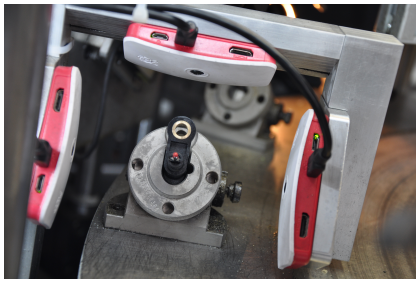

Fig. 1: Prototype camera mount. Each red box contains a Raspberry Pi Zero and a Raspberry Pi v2 camera module.

The check is currently performed by a traditional, non machine learning based, computer vision solution. The piece stops in front of a single camera and is turned by a robotic arm to produce three views. Each of the two soldering seams gets one direct plan view and the last image shows the two bulbs in profile. The last view is necessary as the total width given by the peaks of the soldering bulbs cannot exceed a predefined threshold as to avoid contact with a metal casing that gets attached at a later stage of the production. The legacy solution is not robust and consistent enough. Many pieces are wrongly rejected, which lead to the manufacturer implementing an additional manual check. At the end of each shift, each rejected piece is manually checked and potentially defect-free pieces are re-injected into the production cycle with the legacy quality control inconsistently accepting pieces at the second try. Undetected defects can lead to client rejection or high penalties in case of failure during a warranty period.

The legacy system is to be replaced by a more robust machine learning based solution. In addition, the new system should not only detect defects but also differentiate between 6 defined classes of defects. It was not possible to access the images of the legacy setup easily. Furthermore, the images were very overexposed, as required by the legacy solution, and changes to the lighting parameters were not possible as this would have made continued production impossible. Hence, a new, separate control system has been put in place. The goal of the prototypical setup was to gather image samples and evaluate trained CNN models on the live production. As cloud-computation is not wanted by the manufacturer for security reasons, the prototype should also proof the feasibility of CNN classification inference on the edge. We chose the NVIDIA Jetson Nano [16, 17] as our small factor compute unit. The Nano features a quad-core ARM processor and a GPU with 128 CUDA cores to accelerate the CNN execution with a low power consumption of 10W.

To enable us to take images in the narrow available space, we chose the Raspberry Pi v2 camera module. Three cameras were attached to a support to collect all three views at the same time. The support is shown on Figure 1. The A02 revision carrier board of the Nano features only a single camera

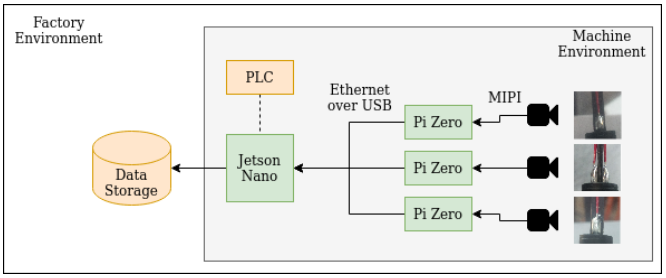

Fig. 2: Schematic prototype setup on the edge. The Jetson Nano reads the PLC to trigger the cameras. Cameras are plugged via MIPI connector to individual Raspberry PI Zeros, which in turn are connected to the Nano via Ethernet over USB. Images and meta data are stored locally.

connector slot, ruling out a direct connection. Instead, each camera was connected to a Raspberry Pi Zero module which in turn was connected via USB to the Nano. Ethernet over USB was used to transmit images in a server-client setup with the Nano as a single client connected to three Pi Zero servers. The camera outputs were cropped to the area of interest before transmitting 480x480 pixel images to the Nano, for reduced latency. To take images at the correct time, a trigger was required to determine when a piece is standing still in front of the cameras. For this, the programmable logic controller (PLC) of the production line was observed, waiting for the signal that is used to trigger the legacy camera system. Finally, the Nano was connected to a local storage to dump samples consisting of the three images as well as additional metadata. Figure 2 illustrates the whole hardware prototype setup.

As classifier, a multi-view CNN (MVCNN) as suggested in [10], was used. The authors suggest to use a single backbone fully-convolutional model to learn and extract features from all given views. The original classifier is replaced by a view-pooling layer and a combination of linear, ReLU and dropout layers. The view pooling layer receives the outputs of the feature extractor, which are max-pooled along the view dimension to allow the classifier to focus on the most salient features. The MVCNN enables to learn dependencies between several 2D views of a 3D object [10]. Furthermore, MVCNN is more resource efficient when compared to using individual CNNs per view, making the solution suitable for computations on the edge with limited resources. The shared model weights reduce the memory requirements significantly. The computation is also more time-efficient as there is no need to switch between models at run-time. Lastly, MVCNN also reduces the labeling effort drastically as only one label per sample is required instead of one label per view. We used the ResNet18 architecture [24] as backbone and chose the small 18 layer variant to enable real-time inference on the Jetson Nano prototype. We followed [10] to remove the original classifier layer and replace it by two blocks of dropout-linear-ReLU, followed by a final linear layer with 7 output neurons, one for each target class of TSMVD.

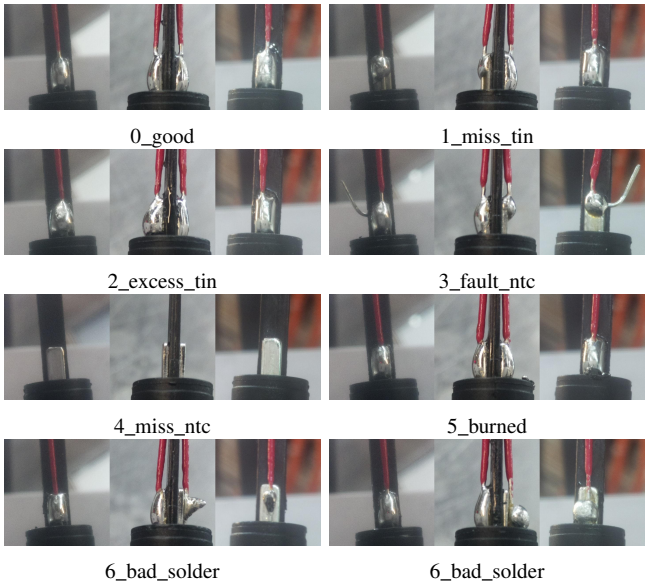

Fig. 3: Examples for each class of the labeled dataset with 3 views each. Two examples provided for class *6_bad_solder*.

Class ID	Description	#Samples
0_good	defect-free piece	437
1_miss_tin	missing tin	270
2_excess_tin	excess of tin	53
3_fault_ntc	faulty NTC	45
4_miss_ntc	missing NTC	35
5_burned	burned support	91
6_bad_solder	bad soldering bulb	91

TABLE I: Distribution of labeled data

IV. TSMVD Dataset

To collect unlabeled data, the prototype was used as described in Section III. The ratio of truly defective pieces on the production line is estimated to be 4%. Training CNNs with extreme class imbalance is, however, still very challenging [8, 9]. We therefore focused on labeling samples of defective pieces to create a more balanced dataset. Over a period of 15 days, a worker randomly selected pieces that were accepted and rejected by the legacy system. On average, 70 pieces were collected per day and an expert from quality assurance sorted through the selection. Afterwards the production was stopped for a short time period and the collected samples were injected into the machine without processing them. This allowed to pass the pieces in front of the hardware prototype and collect the images while knowing the correct label.

Table I provides an overview of the labeled data distribution. The set features 437 good and 585 defective pieces, leading to a ratio of 1.34 in favour of the faulty samples. However, the bad pieces are distributed between 6 classes of defects. The defect types were defined by quality assurance experts and reflect the errors that can be observed on the production line. The differentiation of defects is important as an increase of a particular kind of errors could indicate specific issues with the production line. The goal of the manufacturer is to use this information to detect and solve machine issues faster

and hence improve their overall productivity. Fig. 3 shows one example for each defect type.

The defect-free class makes up for 43.76% of the data. Class *1_miss_tin* is the predominant defect type with 270 samples, 46% of all defective pieces. The defect is identifiable by a lack or complete miss of tin on one or both soldering seams. The tin is applied on a small metal plate which should be completely covered. Without tin, the electrical circuit is not closed making thermal readings impossible. In contrast, samples of class *2_excess_tin* feature an excess of tin. Excessive soldering bulbs could make contact with a metal casing that gets attached in a later production step. This in turn can potentially create a short circuit and lead to unreliable thermal readings or make the unit unusable. Samples of class *3_fault_ntc* are characterized by twisted or badly positioned NTCs. Both ends of the NTC need to be fitted in guiding channels of the support body. When the machine fails to insert the NTC into these channels, the NTC bends away from support. In some cases the NTC is still able to be soldered in the next step but even then the piece needs to be rejected as the loose end of the NTC could touch the metal casing. The top part of the NTC is allowed to be slightly bend as long as it still fits within the casing. Samples of class *4_miss_ntc* are missing the NTC completely. The machine either failed to attach the NTC or it fell of before or even after soldering occurred. It might also be that the metal plates are missing or damaged, making it impossible to solder the NTC. Class *5_burned* is caused by the hot soldering rot touching the support body and melting the plastic. Class *6_bad_solder* combines several types of badly formed soldering bulbs. The most common shape among class *6_bad_solder* are tin peaks, characteristically tall shallow solder points which only cover a small part of the NTC.

In addition to the labeled data, we also collected a large set of 293k unlabeled samples over the span of five months. The minority class *4_miss_ntc* makes up for only 6% of the defects in our labeled data. With an estimated 4% of defective pieces, we could infer a maximum ratio of 0.24% minority samples, or put differently, a minimum imbalance ratio of 1 to 400. However, it is estimated that rare defects only make up for less than 1% and hence the true imbalance ratio on our unlabeled data could be in the magnitude of thousands. An additional challenge of the unlabeled data are unexpected errors such as absent pieces or mistimed camera triggers which lead to blurred images of samples in motion.

The TSMVD dataset offers the opportunity to investigate SSL under real world constraints. It is linked to a domain with clear needs, namely industrial defect detection. In contrast to other publicly available datasets [12, 13] of this domain, it offers a large number of unlabeled samples to explore the potential advantages of SSL.

V. Experiments

The labeled dataset features 1022 samples and a class imbalance ratio of 12.5, with only 35 minority class samples. Furthermore, there is a high visual variance among the samples of a single class of defects. To avoid the risk that a randomly selected test-subset would favour models overfitting on a specific sub-type of defect, we decided to conduct a 4-fold cross validation. In consequence, the samples of each class were randomly split into 4 buckets. Next, 4 training runs were performed per experiment while using 1 bucket for validation and the remaining 3 for training. The training was performed using a NVIDIA Tesla T4.

Method	Recall	Precision	Accuracy
SV	78.34±8.12	87.02±1.18	85.41±3.43
SV-PT	82.20±2.15	87.17±3.48	87.81±1.29
SV-PT-LA	84.29±3.79	88.42±3.94	88.85±2.86
FIX	74.10±4.40	**89.78±2.73**	86.51±2.46
FIX-PT	83.31±2.25	87.43±2.34	88.09±0.91
FIX-PT-LA	85.74±2.66	87.01±2.17	88.63±0.89
DASO	84.27±5.27	84.30±1.37	88.14±1.84
DASO-PT	84.29±3.22	86.24±2.77	88.47±2.26
DASO-PT-LA	**87.83±3.01**	86.40±3.70	**89.24±2.04**

TABLE II: Training results. Values are the mean and standard deviation over 4 cross validation folds. Higher is better, best values are indicated in bold. (SV:=Supervised, FIX:=FixMatch, PT:=Pretrained, LA:=Logit adjustment)

Motivated by the high availability of 293k unlabeled samples, we wanted to investigate if the data can be used to improve models on the validation set. We use FixMatch [7] and DASO [8] for our investigations. DASO extends FixMatch and provides state-of-the-art results for balanced and imbalanced SSL [8]. Furthermore, we evaluate the use of transfer learning under our scarcely labeled setup as well as logit adjustment [19] as a re-balancing technique.

To enable a fair comparison, the following training hyperparameters were determined empirically from initial tests and kept constant for the main experiments. Models were trained for 100 epochs of 500 optimization steps each, for a total of 50k steps. Stochastic gradient decent was used with a batch size of 10, momentum of 0.9, nesterov acceleration enabled and weight decay set to $5\mathrm{e}{-4}$. The learning rate was started at $1\mathrm{e}{-3}$ and reduced smoothly using the cosine annealing scheduler without warm-restarts [25]. For transfer-learning, pretrained ImageNet weights were used for the ResNet18 backbone. As the domain differentiates drastically from ImageNet, all layers were fine-tuned and the starting learning rate reduced to $1\mathrm{e}{-4}$. For the additional FixMatch and DASO hyperparameters we followed prior works on imbalanced SSL [8, 9]. Images were randomly transformed to better reflect the variance that can be observed on the production line. For example, the position of the piece with respect to the camera can change slightly, the color of the materials are subject to change and the illumination is not perfectly controlled with our prototype as we could not fit a box around the camera setup. In addition, FixMatch requires a weak and strong augmentation setting to perform consistency regularization. In their original work, Sohn et. al. use random horizontal flip and crop as weak augmentation and RandAugment [22] for the strong augmentation. In our experiments, RandAugment was used for both weak and strong augmentations. The weak setting was also re-used to augment the labeled samples for supervised and semi-supervised learning. Input images are

Method	Class						
	0_good	1_miss_tin	2_excess_tin	3_fault_ntc	4_miss_ntc	5_burned	6_bad_solder
SV	85.16±5.45	86.98±9.60	89.62±12.43	87.20±10.55	90.35±11.89	**84.224±14.82**	85.58±9.06
SV-PT	87.19±3.72	93.45±2.97	84.81±10.70	86.22±16.78	97.5±5.00	69.76±4.11	91.23±4.18
SV-PT-LA	87.92±3.87	93.61±3.60	79.98±7.98	89.89±10.79	97.75±4.50	78.29±11.53	**91.49±9.99**
FIX	82.75±3.36	92.03±6.53	**91.64±9.73**	90.95±18.10	**100.00±0.00**	82.82±7.73	88.27±11.84
FIX-PT	87.28±4.12	**94.50±1.82**	79.23±7.66	**94.00±9.47**	97.50±5.00	72.29±13.70	87.22±7.88
FIX-PT-LA	90.46±4.17	91.43±1.90	75.06±5.51	91.13±10.34	97.54±4.44	74.82±11.32	88.63±8.90
DASO	90.40±4.60	93.23±1.90	79.47±18.32	68.64±10.51	93.36±8.73	79.65±9.57	85.3±9.90
DASO-PT	89.32±3.83	93.10±3.70	72.54±5..78	83.27±13.23	97.50±5.00	80.82±14.41	87.14±10.19
DASO-PT-LA	**91.52±4.13**	91.87±3.90	74.50±12.23	84.47±11.75	95.72±7.39	77.32±11.58	89.41±7.80

TABLE III: Precision per class. Reported values are the mean and standard deviation over 4 cross validation folds. Higher is better, best values are indicated in bold. (SV:=Supervised, FIX:=FixMatch, PT:=Pretrained, LA:=Logit adjustment)

Method	Class						
	0_good	1_miss_tin	2_excess_tin	3_fault_ntc	4_miss_ntc	5_burned	6_bad_solder
SV	96.24±2.4	93.24±3.85	54.43±15.53	**96.43±7.14**	35.30±7.94	55.69±6.96	84.21±15.43
SV-PT	91.16±4.21	94.86±2.75	73.13±12.26	61.62±18.77	96.43±7.14	68.25±7.49	89.96±5.57
SV-PT-LA	92.66±5.48	94.63±2.37	71.50±11.53	83.47±11.27	96.43±7.14	63.62±10.95	87.70±8.96
FIX	**96.61±2.01**	94.06±4.43	51.51±18.31	33.66±19.28	96.43±7.14	63.62±8.54	82.78±6.91
FIX-PT	91.20±1.91	95.25±2.54	73.32±17.78	58.18±19.05	**100.00±0.00**	68.88±47.33	**93.79±5.40**
FIX-PT-LA	89.97±2.08	94.43±1.31	80.07±15.04	73.98±18.77	96.43±7.14	71.71±17.15	93.58±4.51
DASO	89.83±3.04	**95.63±3.85**	79.88±16.84	64.88±26.46	95.43±7.14	70.31±13.74	92.93±5.66
DASO-PT	91.23±3.12	95.55±3.91	79.74±15.88	67.80±16.21	96.43±7.14	66.75±14.96	92.55±5.04
DASO-PT-LA	90.03±2.68	94.54±2.10	**86.81±11.86**	81.50±21.10	96.43±7.14	**72.47±15.47**	93.05±4.82

TABLE IV: Recall per class. Reported values are the mean and standard deviation over 4 cross validation folds. Higher is better, best values are indicated in bold. (SV:=Supervised, FIX:=FixMatch, PT:=Pretrained, LA:=Logit adjustment)

first randomly flipped horizontally, resized to 224x224px and then further augmented using RandAugment. The difference between the weak and strong settings are the addition of shear, equalize, posterize and solarize as transformation. A full list of hyperparameters can be found online[2].

We report the precision and recall per class in Table III and IV respectively. We take the median over the last 20 training epochs and report the mean and standard deviation over the 4 cross validation folds. Table II shows the accuracy and the arithmetic mean over precision and recall. DASO with pretrained weights and logit adjustment leads to the best recall and overall accuracy, while FixMatch from scratch provides the best precision. Recall is higher for SSL models, indicating the advantage of using the extensive unlabeled data. Transfer leaning improves recall for all methods but leads to a slight drop in precision when using FixMatch. The increase in recall is more significant though, hence we can conclude that pretrained weights can be useful for scarcely labeled data even if a large set of unlabeled data is available. Using logit adjustment further increases the overall performance.

From the per-class details (Table III and IV) we can observe that despite the overall better performance of SSL models, they do struggle on some classes. For example, FixMatch has a particularly low recall on *3_fault_ntc*. The issue is lessened by applying both transfer learning and logit adjustment and the precision on the class is even the highest among all models but the low recall is still problematic. DASO features a relatively low recall on *0_good*, while precision is high. During production this would lead to a lower number of missed errors but also a higher false alarm rate compared to the SV model.

[2]https://tcdata2206.s3.eu-west-3.amazonaws.com/index.html

VI. Conclusion

In this work we discuss the challenges of convolutional neural networks (CNNs) for real world image classification, namely label scarcity and extreme class imbalance. We present a CNN-based defect detection solution for the quality assurance of thermal sensors based on multiple views. We describe our prototypical setup that was used to collect data and evaluate models for real time processing on the edge. Furthermore, we make our TSMVD dataset of 1k labeled and 293k unlabeled samples publicly available to enable researchers to reproduce our results and allow for future work on this challenging task. Motivated by the significantly higher availability of unlabeled data, we conduct an early investigation on using semi-supervised learning (SSL). SSL methods increase the overall results on TSMVD but still struggle on some classes. For future work we plan to explore improvements of the investigated methods towards tasks under the real world constraints presented in this paper.

References

[1] E. N. Malamas, E. G. Petrakis, M. Zervakis, L. Petit, and J.-D. Legat, "A survey on industrial vision systems, applications and tools," *Image and Vision Computing*, vol. 21, no. 2, pp. 171–188, Feb. 2003. [Online]. Available: https://linkinghub.elsevier.com/retrieve/pii/S026288560200152X

[2] O. Russakovsky, J. Deng, H. Su, J. Krause, S. Satheesh, S. Ma, Z. Huang, A. Karpathy, A. Khosla, M. Bernstein, A. C. Berg, and L. Fei-Fei, "ImageNet Large Scale Visual Recognition Challenge," *International Journal of Computer Vision*, vol. 115, no. 3, pp. 211–252,

Dec. 2015. [Online]. Available: http://link.springer.com/10.1007/s11263-015-0816-y

[3] A. Krizhevsky, "Learning Multiple Layers of Features from Tiny Images," 2009.

[4] H. Chen, Y. Pang, Q. Hu, and K. Liu, "Solar cell surface defect inspection based on multispectral convolutional neural network," *Journal of Intelligent Manufacturing*, vol. 31, no. 2, 2020. [Online]. Available: http://link.springer.com/10.1007/s10845-018-1458-z

[5] 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*, vol. 30, no. 6, pp. 2525–2534, Aug. 2019. [Online]. Available: http://link.springer.com/10.1007/s10845-018-1415-x

[6] S. Laine and T. Aila, "Temporal Ensembling for Semi-Supervised Learning," *arXiv preprint arXiv:1610.02242*, 2016. [Online]. Available: http://arxiv.org/abs/1610.02242

[7] K. Sohn, D. Berthelot, N. Carlini, Z. Zhang, H. Zhang, C. A. Raffel, E. D. Cubuk, A. Kurakin, and C.-L. Li, "FixMatch: Simplifying Semi-Supervised Learning with Consistency and Confidence," in *Advances in Neural Information Processing Systems 33*, 2020, pp. 596–608. [Online]. Available: https://proceedings.neurips.cc/paper/2020/hash/06964dce9addb1c5cb5d6e3d9838f733-Abstract.html

[8] Y. Oh, D.-J. Kim, and I. S. Kweon, "DASO: Distribution-Aware Semantics-Oriented Pseudo-Label for Imbalanced Semi-Supervised Learning," in *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, 2022, pp. 9786–9796. [Online]. Available: https://openaccess.thecvf.com/content/CVPR2022/html/Oh_DASO_Distribution-Aware_Semantics-Oriented_Pseudo-Label_for_Imbalanced_Semi-Supervised_Learning_CVPR_2022_paper.html

[9] J. Kim, Y. Hur, S. Park, E. Yang, S. J. Hwang, and J. Shin, "Distribution Aligning Refinery of Pseudo-label for Imbalanced Semi-supervised Learning," *Advances in Neural Information Processing Systems 33*, 2020.

[10] H. Su, S. Maji, E. Kalogerakis, and E. Learned-Miller, "Multi-view Convolutional Neural Networks for 3D Shape Recognition," in *2015 IEEE International Conference on Computer Vision (ICCV)*, Dec. 2015, pp. 945–953. [Online]. Available: http://ieeexplore.ieee.org/document/7410471/

[11] J. R. Ferreira, D. Armando Cardona Cardenas, R. A. Moreno, M. de Fátima de Sá Rebelo, J. E. Krieger, and M. Antonio Gutierrez, "Multi-View Ensemble Convolutional Neural Network to Improve Classification of Pneumonia in Low Contrast Chest X-Ray Images," in *2020 42nd Annual International Conference of the IEEE Engineering in Medicine & Biology Society (EMBC)*, Jul. 2020, pp. 1238–1241.

[12] K. Song, S. Hu, and Y. Yan, "Automatic Recognition of Surface Defects on Hot-rolled Steel Strip using Scattering Convolution Network," *Journal of Computational Information Systems*, vol. 10, no. 7, pp. 3049–3055, 2014.

[13] S. Tang, F. He, X. Huang, and J. Yang, "Online PCB Defect Detector On A New PCB Defect Dataset," *arXiv preprint arXiv:1902.06197*, 2019. [Online]. Available: http://arxiv.org/abs/1902.06197

[14] M. Wieler and T. Hahn, "Weakly supervised learning for industrial optical inspection," in *DAGM symposium in*, 2007.

[15] H. Li, K. Ota, and M. Dong, "Learning IoT in Edge: Deep Learning for the Internet of Things with Edge Computing," *IEEE Network*, vol. 32, no. 1, pp. 96–101, Jan. 2018. [Online]. Available: https://ieeexplore.ieee.org/document/8270639/

[16] S. Mittal, "A Survey on optimized implementation of deep learning models on the NVIDIA Jetson platform," *Journal of Systems Architecture*, vol. 97, pp. 428–442, Aug. 2019. [Online]. Available: https://linkinghub.elsevier.com/retrieve/pii/S1383762118306404

[17] A. A. Suzen, B. Duman, and B. Sen, "Benchmark Analysis of Jetson TX2, Jetson Nano and Raspberry PI using Deep-CNN," in *2020 International Congress on Human-Computer Interaction, Optimization and Robotic Applications (HORA)*, Jun. 2020, pp. 1–5. [Online]. Available: https://ieeexplore.ieee.org/document/9152915/

[18] J. M. Johnson and T. M. Khoshgoftaar, "Survey on deep learning with class imbalance," *Journal of Big Data*, vol. 6, no. 1, p. 27, Mar. 2019. [Online]. Available: https://doi.org/10.1186/s40537-019-0192-5

[19] A. K. Menon, S. Jayasumana, A. S. Rawat, H. Jain, A. Veit, and S. Kumar, "Long-tail learning via logit adjustment," *arXiv preprint arXiv:2007.07314*, 2020. [Online]. Available: http://arxiv.org/abs/2007.07314

[20] S. J. Pan and Q. Yang, "A Survey on Transfer Learning," *IEEE Transactions on Knowledge and Data Engineering*, vol. 22, no. 10, pp. 1345–1359, Oct. 2010. [Online]. Available: http://ieeexplore.ieee.org/document/5288526/

[21] C. Shorten and T. M. Khoshgoftaar, "A survey on Image Data Augmentation for Deep Learning," *Journal of Big Data*, vol. 6, no. 1, pp. 1–48, Dec. 2019. [Online]. Available: https://journalofbigdata.springeropen.com/articles/10.1186/s40537-019-0197-0

[22] E. D. Cubuk, B. Zoph, J. Shlens, and Q. V. Le, "Randaugment: Practical automated data augmentation with a reduced search space," in *2020 IEEE/CVF Conference on Computer Vision and Pattern Recognition Workshops (CVPRW)*, Jun. 2020, pp. 3008–3017. [Online]. Available: https://ieeexplore.ieee.org/document/9150790/

[23] B. Zhang, Y. Wang, W. Hou, H. WU, J. Wang, M. Okumura, and T. Shinozaki, "FlexMatch: Boosting Semi-Supervised Learning with Curriculum Pseudo Labeling," in *Advances in Neural Information Processing Systems 34*, 2021, pp. 18 408–18 419. [Online]. Available: https://proceedings.neurips.cc/paper/2021/hash/

995693c15f439e3d189b06e89d145dd5-Abstract.html

[24] K. He, X. Zhang, S. Ren, and J. Sun, "Deep Residual Learning for Image Recognition," in *2016 IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*, Jun. 2016, pp. 770–778. [Online]. Available: http://ieeexplore.ieee.org/document/7780459/

[25] I. Loshchilov and F. Hutter, "SGDR: Stochastic Gradient Descent with Warm Restarts," *arXiv preprint arXiv:1608.03983*, 2016. [Online]. Available: http://arxiv.org/abs/1608.03983