Learning to Count Arbitrary Industrial Manufacturing Workpieces

Ziyi Jiang, *Student Member, IEEE*, Yikui Zhai, *Senior Member, IEEE*, Feng Ke, *Member, IEEE*, Jianhong Zhou, Angelo Genovese, *Senior Member, IEEE*, Vincenzo Piuri, *Fellow, IEEE*, and Fabio Scotti, *Senior Member, IEEE*

***Abstract*—Man-made workpiece counting is a routine job for manufactory workers; however, this is an error-prone task. In this article, we are interested in detecting and counting arbitrary workpieces in industrial manufacturing. Therefore, we construct a comprehensive and large-scale open-world public benchmark dataset for workpiece counting, called workpiece counting dataset, which includes 121 475 instances of workpieces from 351 different categories. We also propose a novel method for workpiece detection and counting, named two-stage workpiece counting network. The first stage of the network is to develop a class-agnostic detector to localize each workpiece instance, followed by the second stage to employ an unsupervised deep clustering strategy with the backbone network pretrained in a workpiece convolutional autoencoder for decision boundary prediction, achieving workpiece clustering under unknown *K* values. Finally, our experiments show that the proposed method outperforms current mainstream methods, greatly enhancing the efficiency of factory operations.**

***Index Terms*—Counting arbitrary workpiece, two-stage workpiece counting network (TS-WPCNet), workpiece counting dataset (WPCD).**

Manuscript received 10 December 2023; accepted 26 January 2024. This work was supported in part by Guangdong Basic and Applied Basic Research Foundation under Grant 2021A1515011576, in part by Guangdong Science and Technology Planning under Grant 2021A0505030080 and Grant 2021A0505060011, in part by Guangdong Higher Education Innovation and Strengthening School under Grant 2020ZDZX3031, Grant 2022ZDZX1032, and Grant 2023ZDZX1029, in part by Wuyi University Hong Kong and Macao Joint Research and Development Fund under Grant 2022WGALH19, and in part by Guangdong Jiangmen Science and Technology Research under Grant 2220002000246 and Grant 2023760300070008390. Paper no. TII-23-4977. *(Ziyi Jiang and Yikui Zhai contributed equally to this work.) (Corresponding authors: Yikui Zhai; Feng Ke.)*

Ziyi Jiang is with the School of Electronics and Information Engineering, Wuyi University, Jiangmen 529020, China, and also with the School of Electronic and Information Engineering, South China University of Technology, Guangzhou 510641, China (e-mail: zylofor@foxmail.com).

Yikui Zhai and Jianhong Zhou are with the School of Electronics and Information Engineering, Wuyi University, Jiangmen 529020, China (e-mail: yikuizhai@163.com; jackychou_lab@126.com).

Feng Ke is with the School of Electronic and Information Engineering, South China University of Technology, Guangzhou 510641, China (e-mail: fengke@scut.edu.cn).

Angelo Genovese, Vincenzo Piuri, and Fabio Scotti are with the Departimento di Information, Universita, Degli Studi di Milano, 20133 Milano, Italy (e-mail: angelo.genovese@unimi.it; vincenzo.piuri@unimi.it; fabio.scotti@unimi.it).

Data is available online at https://github.com/yikuizhai/WPCD-DATASET.

Color versions of one or more figures in this article are available at https://doi.org/10.1109/TII.2024.3363063.

Digital Object Identifier 10.1109/TII.2024.3363063

I. Introduction

The steel/aluminum (referred to workpiece) industry is the foundation of a country's independent economic system. In recent years, the production of workpieces has been increasing, making manual counting more and more burdensome. A vision-based automated counting process is highly desirable in this regard.

Recently, the importance and applicability of deep learning have been verified, precipitating its application in a growing number of domains [1]. However, vision-based detection algorithms often require plenty of available data samples, which are critical for training an outstanding deep-learning model. For meeting the urgent needs of real industrial scenarios, we established the largest and open-world benchmark workpiece counting dataset, (WPCD), which is novel and comprehensive. Its data were collected from actual industrial manufacturing and included 121 475 objects from 351 classes of workpieces. All images have been annotated with the assistance of experts, and the WPCD dataset can serve as a valuable resource for evaluating new ideas in the industrial vision field. To investigate the contributions of our dataset to real-world workpieces counting applications, we provided consistent annotation for many available instances, template class images, and simple visual samples. The dataset is promising. When factories produce new workpieces, it is usually not necessary for researchers to train their special models with massive ground truth data. If researchers can develop models on their data by adapting our dataset through transfer learning or models trained on our dataset, they will create more effective pipelines and develop more accurate models.

In actual scenarios, accurate counting and detecting workpieces present a significant challenge. This difficulty is reflected in the different levels of this dataset, namely, Level-1, Level-2, and Level-3. Regarding Level-1, the key challenge is teaching models to correctly identify newly produced workpieces in an *open-set detection* task. Collecting and annotating such datasets is expensive and can act as a bottleneck, especially, when dealing with many classes or when the class list is constantly changing in the domain of workpiece counting. In terms of

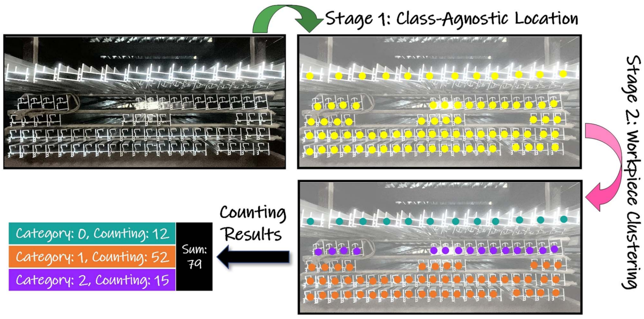

Fig. 1. Workpiece intelligent counting pipeline, the objective is to count the number of each workpiece category and the total number.

Level-2, the challenge lies in how accurately the model detects occluded, ultrasmall, and low-visibility workpieces, and the machine seems to have difficulty locating the positions of these objects. Finally, Level-3 is defined as exploring the essence of the phenomenon. To put it another way, the obstacle is how models parse stacked workpieces into part–whole relationships based on existing categories. In the manufacturing environment, a greater challenge arises when the three levels are put together: how recognition succeeds from such sparse workpieces data and learns rich representations to parse the stacking style of workpieces in a low-visibility environment, referred to as Level-K. For better qualitative analysis, WPCD is further divided into multiple validation sets with different levels, which are validated on different baseline algorithms.

Furthermore, most researchers' work and their own datasets are not publicly available, which poses a challenge for comparing different detection algorithms. To stimulate the development of industrial vision community, we also proposed a novel approach called two-stage workpiece counting network (TS-WPCNet) for workpiece detection and counting. As shown in Fig. 1, the first stage of the network developed a class-agnostic detector to locate the position of each workpiece instance, and the predicted coordinates were then fed into the second stage. This stage introduced an unsupervised deep clustering strategy (UDCS) with a backbone network pretrained in the workpiece convolutional autoencoder (WCAE). Each workpiece instance was encoded to obtain deep semantic information, embedded into a low-dimensional space using the uniform manifold approximation and projection (UMAP), and then using the Dirichlet process mixture (DPM) inference algorithm to predict decision boundary, enabling feature clustering without prior knowledge of the K value. Finally, the statistical results of each workpiece class and overall count were obtained. Fig. 2 depicts the internal structure of an industrial vision counting system in a real environment. Typically, a single system suffices for a workshop, as workers can simply utilize their mobile phones to capture images and perform the counting task. The main contributions of this work are summarized as follows.

1) WPCD, a large-scale open-world benchmark dataset for workpiece detection and counting, has been established. WPCD has been gathered from real-world industrial manufacturing and contained a total of 121 475 objects from 351 distinct categories. As far as we know, it is the first publicly available benchmark that can be used to develop workpiece counting systems.
2) With the objective of learning the distinctive features of each workpiece and effectively filtering out background noise, we have proposed a WCAE. The designed WCAE is superior to generally stacked autoencoders by incorporating spatial and channel relationships between pixels in images.
3) A novel TS-WPCNet has been proposed, which ran on *open-set detection* tasks. The network combined a class-agnostic likelihood module (CALM) with a UDCS, achieving workpiece counting and clustering under unknown K conditions.
4) We empirically evaluated various algorithms in the benchmark to serve baselines and provided a roadmap for future research. Experimental results show that our method surpasses current mainstream methods.

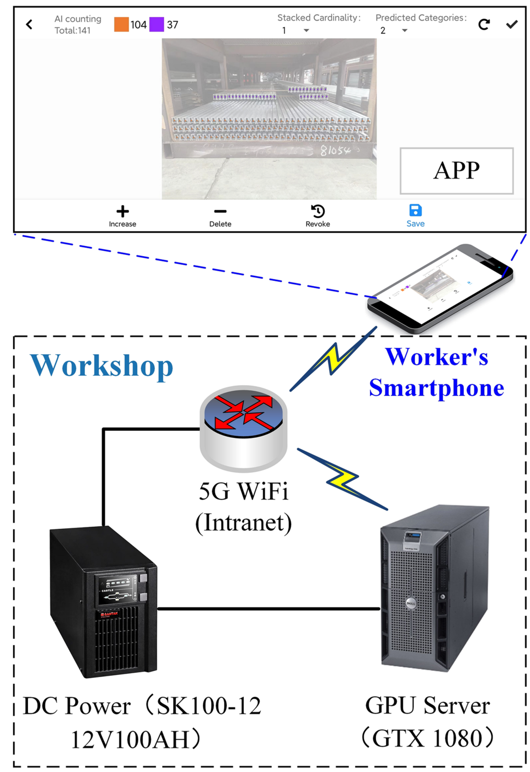

Fig. 2. Internal structure of industrial visual counting system, and TS-WPCNet is deployed in GPU server. Workers can simply utilize their smartphones to capture images and perform the counting.

The rest of this article is organized as follows. Section II introduces the related works. Section III presents an overview of the WPCD. Section IV details the TS-WPCNet framework, and the experimental results are explained in Section V. Finally, Section VI concludes this article.

II. Related Work

This section gives an overview of current workpiece counting development in manufacturing and presents commonly used industrial and benchmark datasets.

TABLE I
KEY PROPERTIES FOR RELATED BENCHMARKS

Dataset	Total	Categories	Average Object/Image	Tasks	Resolution	Annotation	Few shot	Dense	Stack	#BB	Level-K
Surface Defect [18]	1800	6	1	Classification	200×200	-	✗	✗	✗	✗	✗
MVTec AD [19]	5354	73	1	Classification	700∼1024	-	✗	✗	✗	✗	✗
Pascal VOC [11]	22531	20	2.71	Detection	71∼500	Box	✗	✗	✗	✓	✗
MS COCO [12]	**328000**	80	7.7	Detection	51∼640	Box	✗	✗	✗	✓	✗
Paste-f [14]	91	259	53.5	Detection	956∼1280	Box	✓	✓	✓	✓	✗
CARPK [13]	1448	1	62	Counting	1280×1280	Box	✗	✓	✗	✓	✗
Penguins [16]	82000	1	25	Counting	1000∼2500	Point	✗	✓	✗	✗	✗
WPCD(Ours)	1029	**351**	**118**	Counting	1080∼4608	Polygon	✓	✓	✓	✓	✓

#BB: Bounding box labels available for measuring counting Accuracy?
The bold values represent the largest numbers.

A. Different Methods of Counting Workpieces

Workpiece counting is an important step of the workpiece production and sales process. When workpieces are bundled, traditional counting relies on workers marking the workpiece with different colored pens to distinguish between counted and uncounted workpieces. Manual counting is susceptible to inaccuracies due to visual fatigue and memory confusion, resulting in reduced accuracy and potentially significant economic loss. With regards to this, some professionals use traditional image processing methods to count workpieces. For example, researchers have utilized the Hough transform to identify connected regions of the image representing workpieces, or employed the least squares method to fit ellipses for the detection of round workpieces [2], [3]. In a workpiece classification system, Wang et al. [4] proposed an object detection method based on pixel intensity comparison and conducted experiments using nuts and gears as the objects of interest, achieving efficient detection. However, these methods are limited to counting workpieces with distinct categories, clearly defined contours, and organized arrangements.

The automatic counting of workpieces was first achieved by using electromechanical equipment [5]. The equipment lays workpieces on the conveyor belt and uses the occlusion pulse to implement automatic counting through the photoelectric sensor. Recently, visual-based methods for automatic object counting have attracted considerable attention. Jiang et al. [6] proposed a method for contour segmentation to achieve workpiece counting, but it may consume all computational resources and fail to accomplish the task when dealing with hundreds or thousands of crowded instances. In scenarios where workpiece stacks are heavily crowded, density estimation methods [7], [8] are commonly employed as a solution. These regression-based techniques, however, often fail to provide accurate workpiece localization due to two primary causes: 1) the density map comprises indistinctive Gaussian blobs, and 2) significant overlaps occur within the dense density map region. Methods based on object detection can also be used to achieve counting tasks, but they are limited to fixed categories. If there are too many categories or if the categories continue to increase (e.g., 5% of new workpieces can be produced per month), ordinary detectors will become ineffective. TS-WPCNet, through its concept of class-agnostic localization, enables it to handle the majority of scenarios and effectively addresses the challenges caused by the continuous increase in categories, providing a strong solution for practical applications.

B. Comparison of Benchmark Datasets

Existing research has seldom addressed the challenges in the workpiece counting domain. Many benchmarks were purely designed for validating standard object detection, or crowd-counting algorithms. We have investigated a few datasets related to industrial and counting scenarios, and summarized the key differences in Table I. Pascal VOC [9] and MS COCO [10] are commonly used in object detection challenges, but neither of them provides scenes with densely stacked items. Although CARPK [11], Paste-f [12], and Penguins [13] perform the counting function of complex scenes, they are far away from the open-set counting paradigm. In the industrial field, a comprehensive dataset for real-world workpiece recognition scenarios in an open-world environment, such as our WPCD dataset, is currently lacking. The existing datasets in this domain mostly focus on defect detection [14], [15], with a limited scope of a few categories and a small amount of data, hindering the evaluation of new algorithms' generalizability. Although Bergmann et al. [15] included multiple categories of samples, it is artificially generated and not representative of real industrial data. As evident from Table I, our newly established WPCD dataset, described in Section III, provides one to three orders of magnitude more objects per image than almost all these benchmarks and offers two to three orders of magnitude fewer images than most datasets. Furthermore, the categories of the WPCD are continually updated (5% of new workpieces can be produced each month), implying that it has difficulties in collecting and maintaining a dataset.

III. WPCD BENCHMARK

In this section, we will detail the WPCD. By presenting these details, we aim to give researchers a clearer picture of the dataset and its capabilities.

A. Data Description

The WPCD dataset comprises 1029 high-resolution images (1080∼4608), collected by workers using their personal smartphones. These images are accompanied by comprehensive annotations, with each workpiece assigned its own polygon region, and only workpieces that are identical are grouped into the same class. In addition, each object was assigned the "difficult" flag when it was difficult for the annotator to assign a class without guessing. We ended up having 121 475 objects of 351

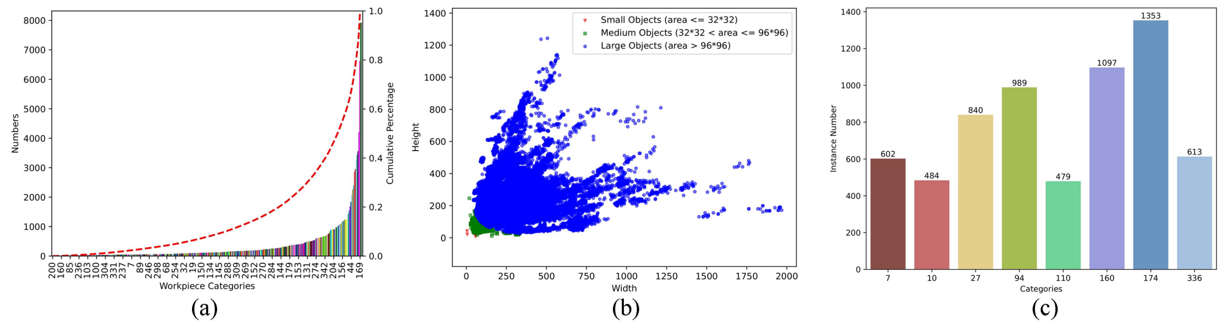

Fig. 3. Visualization of the WPCD attributes. (a) Number of instances in each workpiece category, with the cumulative distribution function shown as a red dashed line. (b) Scatter plot of the width and height of all targets. (c) Eight most common workpiece categories used to evaluate clustering algorithms, and the number of instances in each category.

classes annotated. Meanwhile, a database of template classes is created, where representative instances of each workpiece class are selected as supportive features. Based on the typical workflow, it is common practice to group workpieces of the same or similar categories in one area to facilitate the statistics.

B. Data Annotation and Analysis

We invited professionals to select some common and representative classes of workpieces as a baseline (351 categories were selected), and then, around 30 engineers were involved in collecting and annotating the data. First, template classes were provided by engineers intercepting the cross-sectional drawings of each class, and they were named in order with Arabic numerals (too many classes). Second, VGG image annotator was used to mark the contour of the workpiece in each image, and the annotated region attributes were named according to the template class name. Finally, 121 475 objects in 1029 images had been marked. Fig. 3(a) presents the overall distribution of the number of workpieces, with the highest category reaching 7925 instances and the lowest category having only three instances, presenting the problem of extreme category imbalance. This also indicates that conventional class-related detectors would fail in this challenge. Moreover, Fig. 3(b) visualizes the minimum outer rectangular box width and height corresponding to the outline of all objects.

C. Counting Region Analysis and Data Splits

WPCD can be employed in both fully supervised and few-shot learning settings. Specifically, the template images serve as *support* whereas the count images are treated as *query* images when used in training. To assess the efficacy of our proposed method in diverse scenarios, we conducted validates on 36 388 objects in 152 classes of workpieces (called Level-K), where 311 images of workpieces were selected, and the corresponding images were excluded from the training set. Furthermore, we systematically classified them into three difficulty levels (Level-1–Level-3) for comprehensive method evaluation. The Level-1 set consists of sparsely distributed workpieces with clearly defined relationships and high visibility. It includes 120 images with 6635 objects from 78 classes of workpieces. The Level-2 set is characterized by dense arrangement, a large number of workpieces, low visibility, and occlusions. It includes 98 images with 17 795 objects from 41 classes of workpieces. The Level-3 set was designed to be highly challenging, with almost all workpieces stacked, making the counting task more difficult. It contains 100 images with 13 200 objects from 40 classes of workpieces. It is worth noting that they are not orthogonal because a few images satisfy different levels simultaneously, but the categories of the train set and each level set are orthogonal. The divided results are displayed in Table II.

TABLE II
WPCD IS DIVIDED INTO TRAIN SET AND MULTIPLE VALIDATION SETS

Property	Train	Level-1	Level-2	Level-3	Level-K
Objects	85 087	6635	17 795	13 200	36 388
Images	718	120	98	100	311
Categories	199	78	41	40	152

After constructing the WPCD, we selected the top eight most abundant and frequently occurring categories of workpieces (7, 10, 27, 94, 110, 160, 174, and 336) from Level-1 and Level-2 to evaluate our clustering algorithm, in line with real-world scenarios. The number of workpieces in each category is shown in Fig. 3(c). We did not select workpieces from Level-3 because of the occurrence of single or multiple targets for the same category of workpieces that were stacked on top of each other, which would affect performance evaluation. In addition, to avoid clustering performance bias due to extremely imbalanced classes, we excluded highly similar images from the categories with numerous instances. Finally, we obtained a balanced subset of eight categories containing 6457 images for clustering evaluation.

IV. PROPOSED METHOD

In this section, a novel TS-WPCNet is presented. Fig. 4 depicts the overall architecture of TS-WPCNet.

A. Class-Agnostic Likelihood Module

The goal of a class-agnostic object detector is to accurately predict the position $b_j \in \mathbb{R}^4$ of all foreground objects j and their corresponding likelihood score $s_j \in \mathbb{R}^2$ in an image.

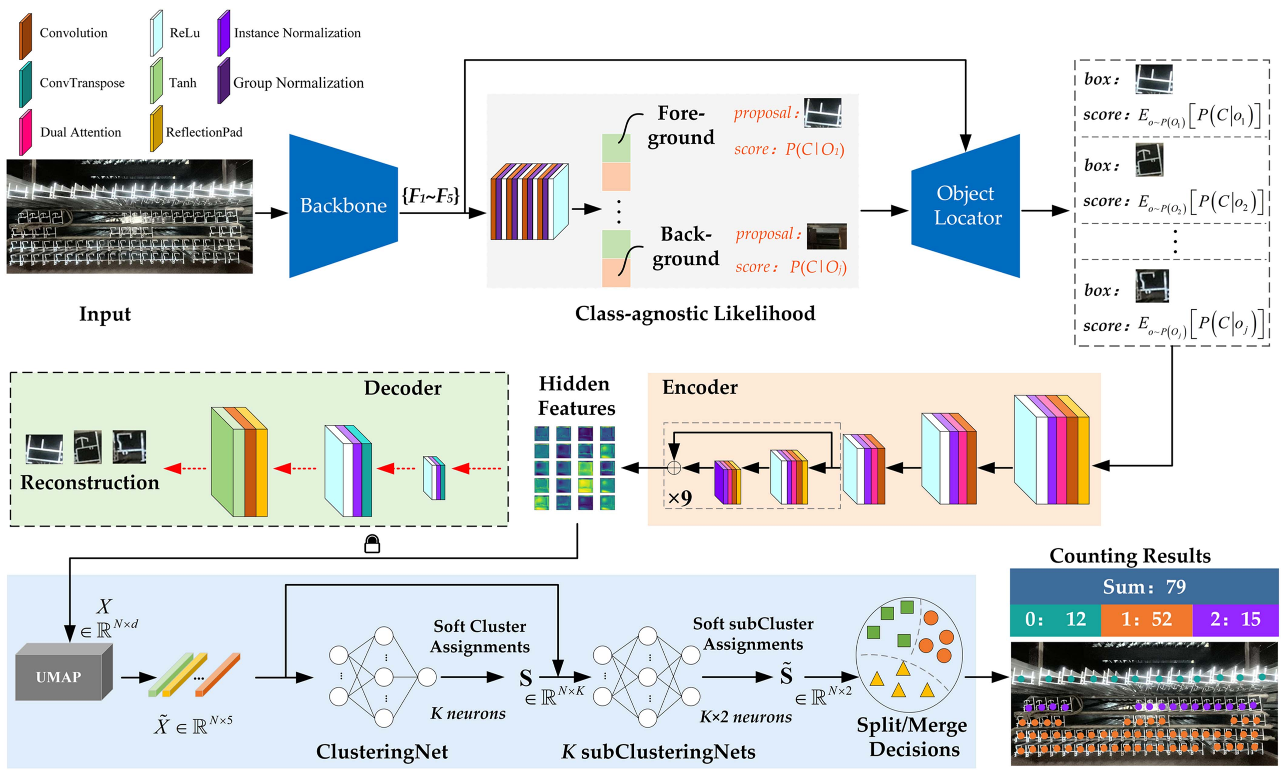

Fig. 4. *TS-WPCNet:* The network aims to detect and count arbitrary workpieces in industrial manufacturing. In the first stage, a class-agnostic detector is designed by incorporating the CALM module to accurately locate each instance of the workpieces. In the second stage, a UDCS is introduced to achieve workpiece clustering under unknown K conditions, with the backbone network pretrained on WCAE. Finally, the statistical results of each workpiece class and overall count are obtained.

The worker captured the image $I \in \mathbb{R}^{H\times W\times C}$ using mobile terminal and transferred it to our algorithm server. Our objective is to generate a set J of bounding boxes $\{b_1, \ldots b_j\}$ and their associated class distribution $d_j(c) = P(C_j = c)$, where c indicates whether the object belongs to the foreground target. In this work, we propose a CALM that focuses exclusively on determining the distribution of probabilities between foreground and background, as shown in Fig. 4 (top). We accomplish this without making any substantial modifications to the feature extraction network and regression heads of traditional detectors. We continued to utilize FPN-ResNet50 [16] as the backbone to generate feature maps at different levels $\{F_1 \sim F_5\}$, and employed a cascaded region-based convolutional neural networks (RCNN) [17] to refine the location of each foreground object. In order to enhance the model's representational capacity, a dual attention mechanism [18] was incorporated into the backbone to dynamically select region of interest from the feature maps.

During the training, a foreground label of 1 was initially assigned to all workpiece categories, and any unassigned objects were labeled as background with 0 (without requiring knowledge of the specific workpiece category). Assuming that there are n samples with foreground labels, and since the samples are independent of each other, the maximum likelihood optimization function can be constructed as

$$L_{\text{fg}-\text{bg}} = \arg\max_{\boldsymbol{\theta}} \prod_{j=1}^{n} P(C_j, O_j; \boldsymbol{\theta}) \tag{1}$$

where $\boldsymbol{\theta}$ denotes the model parameters, and C_j and O_j represent the foreground label and the predicted foreground label during the generation of proposals, respectively. For the jth background sample, $P(\text{bg}|O_j = 0) = 1$. Thus, the logarithmic expansion of (1) can be expressed as

$$\begin{aligned} L_{\text{fg}-\text{bg}} = {} & \arg\max_{\boldsymbol{\theta}} \sum_{j=1}^{n} \left[\log P\left(\text{fg} \mid O_j = 1\right) + \log P\left(O_j = 1\right)\right] \\ & + \arg\max_{\boldsymbol{\theta}} \sum_{j=1}^{n} \log \left[P\left(\text{bg} \mid O_j = 1\right) P\left(O_j = 1\right)\right. \\ & \left. + P\left(O_j = 0\right)\right]. \end{aligned} \tag{2}$$

It is important to note that $P(O_j = 0)$ is calculated during the generation of the proposal, whereas $P(\text{bg} \mid O_j = 1)$ is calculated in the object locator. Combining the loss and gradient calculations for both may lead to the evaluation of all previous outputs for later outputs, making training more difficult. To address this concern, Zhou et al. [19] proposed deriving lower bounds for $P(\text{bg})$ by applying Jensen's inequality and leveraging the monotonically increasing property of $P(\text{bg} \mid O_j = 1)P(O_j = 1) \geq 0$. The resulting optimized likelihood function can be expressed by the following equation:

$$L_{\text{fg}-\text{bg}} = \arg\max_{\boldsymbol{\theta}} \sum_{j=1}^{n} \left[\log P\left(\text{fg} \mid O_j = 1\right) + \log P\left(O_j = 1\right)\right]$$

$$+\arg\max_{\boldsymbol{\theta}} \sum_{j=1}^{n} P(O_j = 1) \log P(\text{bg} \mid O_j = 1)$$
$$+\arg\max_{\boldsymbol{\theta}} \sum_{j=1}^{n} P(O_j = 0). \quad (3)$$

In other words, the feature maps f from $\{F_1 \sim F_5\}$ are fed into the proposal generator G to calculate the foreground likelihood $P(C|O_j)$ for each proposal. Subsequently, this likelihood score is used as the prior probability distribution of the foreground for the object locator to refine the prediction score $s_j = E_{o\sim P(O_j)}[P(C|o_j)]$.

B. Workpiece Convolutional Autoencoder

As previously mentioned, each workpiece instance in the image was obtained during the first stage, but their specific categories are unknown. To extract the category features of these instances and filter out noise before using the UDCS, a WCAE was designed. The WCAE follows an asymmetric encoder–decoder structure, as depicted in Fig. 4 (middle). By controlling the dimension of the hidden feature to be smaller than that of the input data, the WCAE forced the autoencoder to capture the most salient features of the data and avoid degenerate solutions caused by mode collapse. Experimental results demonstrate that the proposed WCAE design produces compact representations and is effective for subsequent workpiece clustering. The parameters of the encoder E_ω and decoder $D_{\omega'}$ were updated by minimizing the reconstruction loss

$$L_r = \frac{1}{N}\sum_{i=1}^{N} \|D_{\omega'}(E_\omega(b_j)) - b_j\|_2^2 \quad (4)$$

where N is the number of instances, and $b_j \in \mathbb{R}^{H\times W\times 3}$ is the jth instance. Our work applied the Huber loss to the image reconstruction training of WCAE.

C. Unsupervised Deep Clustering Strategy

Assuming that $J = (x_i)_{i=1}^{N}$ represents a set of sample points from the first stage, it is necessary to divide J into K distinct and orthogonal groups. We use p_i to denote the assignment of point x_i to a cluster when the clustering labels for x_i are known. Cluster k consists of all points labeled as k, that is, $(x_i)_{i:p_{i=k}}$.

The Dirichlet process Gaussian mixture model is capable of modeling infinitely many Gaussian distributions

$$p(x \mid (\mu_l, \Sigma_l, \pi_l)_{l=1}^{\infty}) = \sum_{l=1}^{\infty} \pi_l N(x; \mu_l, \Sigma_l). \quad (5)$$

Here, $\mu_l \in \mathbb{R}^d$ and a d-by-d Σ_l represent the mean and covariance matrix, respectively, of the Gaussian probability density function N evaluated at $x \in \mathbb{R}^d$, with $\pi_l > 0$ and $\sum_{l=1}^{\infty}\pi_l = 1$. The UDCS consists of the UMAP dimension reduction algorithm, a clustering network, and K subclustering networks (each assigned to a k, $k \in \{1,\ldots,K\}$), as shown in Fig. 4 (bottom). For avoiding a dimensional curse, the UMAP was utilized to map high-dimensional data in the original embedding space to a low-dimensional space, i.e., $\{J \in \mathbb{R}^d \to \tilde{J} \in \mathbb{R}^5\}$. Given low-dimensional features $\tilde{J}$ and an initial K, the clustering network F_1 generates K soft labels for each sample by

$$F_1(\tilde{X}) = \mathbf{S} = \{\mathbf{s}_i\}_{i=1}^{N} = \left\{(s_{i,k})_{i=1}^{N}\right\}_{k=1}^{K} \quad (6)$$

where $s_{i,k} \in [0,1]$ is the soft label of x_i to cluster k, and $\sum_{k=1}^{K} s_{i,k} = 1$. Subsequently, $p_i = \arg\max_k s_{i,k}$ was used to compute the hard label $\{(p_i)_{i=1}^{N}\}$ for each sample. After obtaining the initial cluster labels for each sample, each subclustering network F_{sub}^k was assigned to the sample cluster that has been clustered by hard labels, and further generated soft subcluster labels to propose cluster splitting or merging

$$F_{\text{sub}}^k(\tilde{X}) = \tilde{\mathbf{S}}_k = \{\tilde{\mathbf{s}}_i\}_{i:p_i=k} = \left\{(\tilde{s}_{i,j})_{j=1}^{2}\right\}_{i:p_i=k} \quad (7)$$

where $\tilde{s}_{i,j} \in [0,1]$ is the soft label of x_i to subcluster j ($j \in [1,2]$), and $\tilde{s}_{i,1} + \tilde{s}_{i,2} = 1\ \forall k \in \{1,\ldots K\}$.

According to [20] and [21], the split/merge model utilizes two explicit subclusters to augment each conventional cluster and assigns a subcluster label, $\tilde{p}_i \in \{1,2\}$, to each sample point to indicate whether it belongs to the left or right subcluster, each subcluster with an associated weight $\tilde{\pi}_k = \{\tilde{\pi}_{k,1}, \tilde{\pi}_{k,2}\}$ and parameter $\tilde{\vartheta}_k = \{\tilde{\vartheta}_{k,1}, \tilde{\vartheta}_{k,2}\}$. During the inference process, the network produces proposals to split cluster k into subclusters at fixed epochs. The acceptance probability of this split is represented by $\min(1, H_s)$, and H_s is the "Hastings ratio." Upon acceptance of a split proposal, K is updated, and the units in the final layer of the clustering and subclustering network will increase accordingly. In order to incorporate the network changes, we will duplicate the kth neuron in the output layer of the clustering network along with all the weights connected to the preceding hidden layer. The parameters learned by the subclustering network will then be utilized to initialize the parameters of these two new clusters. Like the splitting process, the merging proposals use the Hastings ratio, $H_m = 1H_s$, to accept or reject them. However, it is important to note that the decision will not accept a proposal that merges a cluster with multiple clusters at the same time. To avoid this, only the three nearest neighbors are considered when merging each cluster. In contrast to splitting, when a merge proposal is accepted, the value of K is decreased. Therefore, the last merged cluster unit and all the weights connecting to the previously hidden layer are removed, and a new cluster unit is then reinitialized.

Finally, for each data x_i and each cluster k, $\mathbf{s}_i^E = (s_{i,k}^E)_{k=1}^{K}$ is used to estimate the posterior probability in the E-step, where

$$s_{i,k}^E = \frac{\pi_k N(x_i; \mu_k, \Sigma_k)}{\sum_{k'=1}^{K} \pi_{k'} N(x_i; \mu_{k'}, \Sigma_{k'})} \quad (8)$$

is computed using $(\pi_k, \mu_k, \Sigma_k)_{k=1}^{K}$ from the previous epoch, and $\sum_{k=1}^{K} s_{i,k}^E = 1$. The clustering network utilizes Kullback–Leibler divergence loss to update its parameters. Afterward, an M-step is executed after every epoch to reestimate each cluster. For the subclustering network, the isotropic loss [20] is used to optimize network parameters

$$L_{\text{sub}} = \sum_{k=1}^{K}\sum_{i=1}^{N_k}\sum_{j=1}^{2} \tilde{s}_{i,j} \|x_i - \tilde{\mu}_{k,j}\|_2^2. \quad (9)$$

Here, $\tilde{\mu}_{k,j}$ is the mean of the jth subcluster of cluster k.

V. Experimental Results and Analysis

We first provided empirical evaluations of various algorithms in the benchmark to establish baselines, and then evaluated the performance of TS-WPCNet in real-world scenarios.

A. Experimental Settings

Our class-agnostic locator was constructed in the Detectron2. We utilized a transfer learning strategy by loading the pretrained ResNet50 on ImageNet into the backbone network and initialized the remaining new convolutional layers with *Kaiming normalization*. During training, the batch size and iteration number were set to 16 and 15 000, respectively. The optimizer was set to stochastic gradient descent (SGD), the weight decay was set to $1e^{-4}$, the momentum was 0.934, and the learning rate was initialized at $1.25e^{-3}$ using a linear warmup strategy, then reduced according to the milestones [6000, 9000, 12 000]. Furthermore, the average values and variances of the RGB channels in the image dataset were [0.381, 0.349, 0.331] and [0.255, 0.245, 0.235], respectively. To ensure fair comparison, all other parameters of the models were adjusted to the best and were optimized using a linear ramp-up learning rate. For WCAE pretraining, each image in the dataset of 121 475 instances was initially resized to 112×112 and normalized with mean and variance of [0.5, 0.5, 0.5]. We set the batch size and epoch to 256 and 400, respectively. For the UDCS inference algorithm, we set the hidden layers of both the clustering network and subclustering network to 50, and initialized K value to 1 (as most workpiece areas have only one class). The batch size and epoch were set to 512 and 400, and the learning rates of the clustering network and subclustering network were set to $2.5e^{-4}$ and $5e^{-4}$. The α and ε in H_s were set to 5 and $5e^{-2}$, respectively, and they started proposing splitting or merging after 55 epochs and performed them every ten epochs.

The experiments were all conducted on Ubuntu 18.04 operating system using PyTorch 1.8.0, torchvision 0.9.1, cudatoolkit 10.2 with an Nvidia Titan RTX (48 G memory), and inference was performed on ground truth (GTX) 1080 8 G GPU in actual scenarios.

B. Evaluation Metrics

Workers exhibit heightened concern regarding occurrences of false positives and false negatives within a given image. This concern stems from the subsequent workflow steps, which involve manually correcting the count of workpieces within the identified region. Notably, upon practical deployment of the system, the need for differentiation based on level ranking becomes unnecessary. Consequently, only the overall false positive rate and false negative rate of the algorithm require consideration. Thus, we proposed a workpiece counting error. It calculated the error by enumerating the number of false negatives and positives, as following:

$$W^E = \frac{1}{N}\sum_{n=1}^{N}\left(|\mathrm{FN}_n| + |\mathrm{FP}_n|\right) \tag{10}$$

where N is the total number of images, and FN_n and FP_n are, respectively, the sets of false negatives and positives in each validating image. The performance of object detection is measured by Pascal VOC metric [11], which is the mean average precision, mAP, at the intersection-over-union threshold of 0.5. The metric proficiently disregards certain detections through the utilization of the “difficult” flag.

The performance of clustering algorithms is typically evaluated based on three common metrics: clustering accuracy (Acc), normalized mutual information (NMI), and adjusted rand index (ARI). When ground truth labels are not available, this work utilizes the Silhouette coefficient (SC) and Calinski–Harabasz Index (CH) for evaluation purposes.

TABLE III
mAP Comparison of Mainstream Object Locator on WPCD, Without Assigning Specific Class Labels

Methods	Level-1	Level-2	Level-3	Level-K
YOLOX [22]	0.691	0.420	0.564	0.517
YOLO-R [23]	0.367	0.320	0.379	0.334
YOLOv7 [24]	0.170	0.100	0.044	0.253
Faster RCNN [25]	0.704	0.414	0.543	0.522
Mask RCNN [26]	**0.827**	0.406	0.502	0.519
SOLOv2 [27]	0.799	0.262	0.251	0.378
BlendMask [28]	0.661	0.423	0.558	0.517
BoxInst [29]	0.822	0.375	0.200	0.299
CondInst [30]	0.822	0.363	0.471	0.493
DETR [31]	0.661	0.305	0.440	0.416
VitDet [32]	0.723	0.429	0.564	0.523
Ours	0.821	**0.475**	**0.634**	**0.594**

The bold values represent state-of-the-art performance.

C. Algorithm Benchmarking on WPCD

1) Benchmarking for Location-Based Methods: As of now, there is extensive research on object detection in deep learning, and mature algorithms include conventional locators, such as YOLOX [22], YOLO-R [23], YOLOv7 [24], Faster RCNN [25], Mask RCNN [26], SOLOv2 [27], BlendMask [28], BoxInst [29], CondInst [30], as well as Transformer detectors, such as DETR [31] and VitDet [32], which have achieved excellent performance in various benchmarks. Therefore, we compared our proposed method with these models to jointly validate the effectiveness of WPCD and provide benchmark results. However, these general locators are limited to certain categories, which make them impractical for counting arbitrary workpiece categories. Consequently, in order to simulate real-world production environments, benchmarks for these detectors do not assign specific category labels. Rather, as mentioned earlier, all objects are only assigned foreground label. Table III presents a comparison of our method with mainstream object location algorithms on the WPCD dataset. The experiments indicate that the proposed algorithm exhibits significant advantages in the mAP across Level-1–Level-3, with results of 82.1%, 47.5%, and 63.4% respectively. The visualization of object counting results

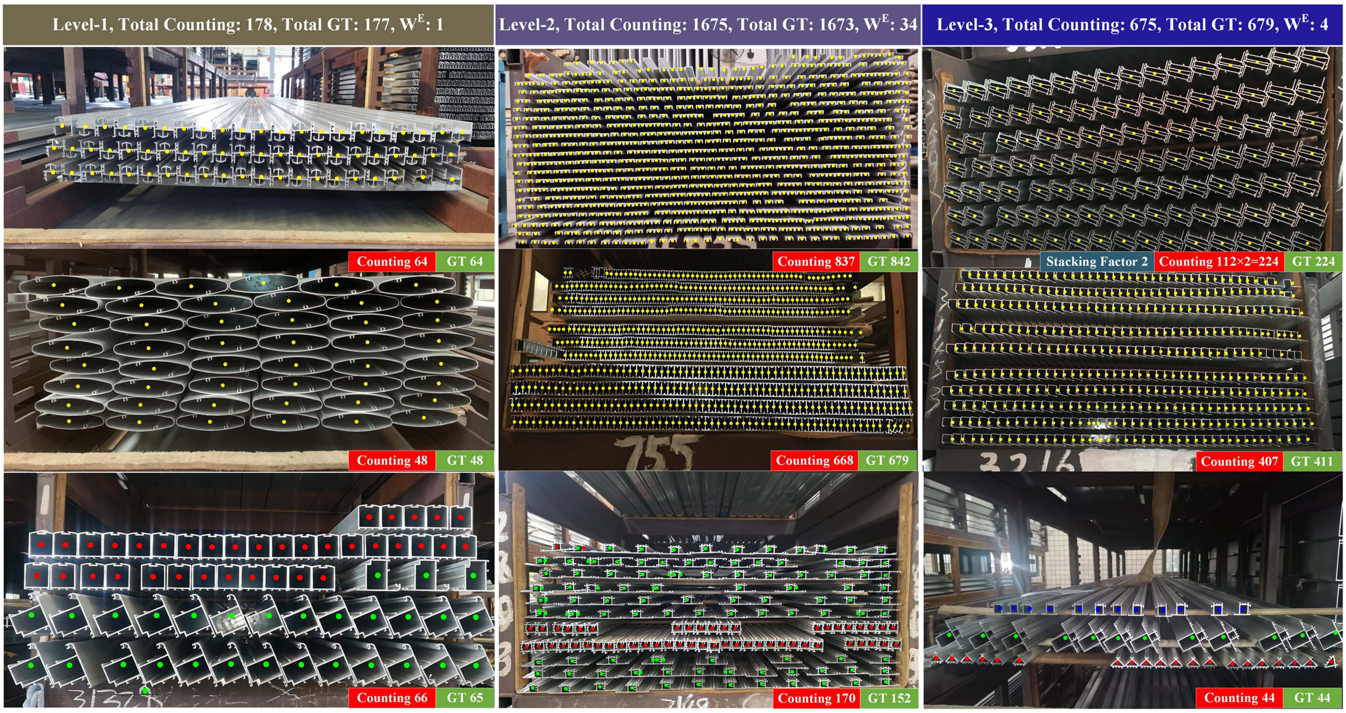

Fig. 5. Proposed method provides visual counting results from Level-1 to Level-3 (the bounding box is replaced by its center point).

TABLE IV
KEY COMPONENTS AFFECTING DETECTOR W^E PERFORMANCE

Methods	Level-1	Level-2	Level-3	Level-K
Baseline [19]	37.6	101.5	54.4	67.7
+CALM (Ours)	11.8	98.2	50.7	50.1
+ CALM+Attention (Ours)	**11.2**	**97.7**	**49.8**	**49.3**

The bold values represent state-of-the-art performance.

TABLE V
W^E COMPARISON OF MAINSTREAM FSOD ON WPCD

Methods	Training Data	Level-1	Level-2	Level-3	Level-K
TFA [33]	1-shot	54.3	178.9	129.3	114.9
TFA [33]	5-shot	50.2	180.3	128.5	113.5
TFA [33]	10-shot	44.4	177.3	125.6	109.5
FSCE [34]	1-shot	49.9	175.9	129.6	112.4
FSCE [34]	5-shot	37.1	151.6	108.4	91.8
FSCE [34]	10-shot	**33.7**	134.1	96.9	83.8
OS2D-s [12]	1-shot	184.3	282.2	228.1	811.2
OS2D-m [12]	1-shot	48.9	**97.4**	**61.4**	**79.2**

The bold values represent state-of-the-art performance.

is presented in Fig. 5 (top and middle). Also, Table IV displays the impact of the key components of the proposed detector on the workpiece error performance. It is worth noting that it is currently difficult to perfectly resolve subparts of stacked objects. Due to the typical workflow, most stacked objects in current real-world scenarios are placed as bundled objects, meaning that the number of stacked objects in each bundle is fixed. To address this, this article first recognized multiple stacked objects as a single overall target, and then multiplied the number of recognized targets by the stacking factor to obtain the final counting result, as shown in Fig. 5 (upper right-hand side corner). During the algorithm verification process, recognition of one bundle of objects was considered correct. Algorithms based on the standard fully supervised paradigm found it difficult to collect and maintain a dataset that had reached a new level, and the continuous addition of object categories would make the model face the dilemma of slow forgetting of old knowledge and the need to quickly adapt to new knowledge. Fortunately, when disregarding class division, the proposed class-agnostic detector can handle most *unseen class*, and based on feedback from practical applications, this benchmark has met the requirements of most scenarios.

2) Benchmarking for Few-Shot Paradigm Object Detector (FSOD) Methods: As mentioned earlier, WPCD also supports few-shot object detection (FSOD)/one-shot paradigm object detectors, which allows professionals in future work to directly extract several template categories from the original image and calculate the number of restricted workpiece areas. We conducted baseline tests on three of the most popular few-shot object detection models, TFA [33], FSCE [34], and OS2D [12], at different shots, and used these results as benchmarks for future research. It is worth mentioning that OS2D is a one-shot detector, and it only needs to understand a single demonstration of an object class given by the inference phase and make a detection. Table V presents the baseline results of these methods on the WPCD benchmark. The FSOD baseline performs much worse than the performance of standard object detectors, indicating the difficulty of the few-shot paradigm. As an ablation study, the experimental results of 1-shot, 5-shot, and 10-shot are reported in the table. These models create numerous proposal boxes

TABLE VI
W^E COMPARISON OF THE CLASS-AGNOSTIC COUNTER ON WPCD

Methods	Label	Level-1	Level-2	Level-3	Level-K	Time (fps)
BMNet [35]	point	506.6	2263.2	1515.3	1335.1	2.36
FIDTM [7]	point	217.2	805.6	394.5	458.8	1.05
FamNet [36]	point	52.3	179.4	130.0	114.1	2.18
CAN [37]	point	28.4	108.9	64.5	63.4	0.48
CSRNet [38]	point	**18.1**	**100.9**	**50.4**	**60.6**	0.70
TopoCount [39]	point	77.6	247.7	157.2	169.0	0.32
DMCount [40]	point	48.8	127.5	88.6	84.6	1.22
STEERER [41]	point	50.6	115.3	67.1	75.2	**11.53**
YOLOX [22]	box	70.8	123.8	74.6	88.1	4.59
YOLO-R [23]	box	286.9	340.6	304.8	307.0	7.14
YOLOv7 [24]	box	292.9	401.8	390.4	134.0	2.62
FasterRCNN [25]	box	26.9	115.7	73.3	65.0	1.07
DETR [31]	box	73.1	146.8	92.7	101.4	1.08
VitDet [32]	box	30.5	114.3	68.3	67.7	2.59
Ours	box	**11.2**	**97.7**	**49.8**	**49.3**	1.17

The bold values represent state-of-the-art performance.

for improving the recall rate, which leads to many redundant detection boxes during actual detection and a sharp increase in the counting error of objects. Surprisingly, the performance of OS2D exhibits significant errors in single-scale training, while achieving a level comparable with that of general detectors in multiscale training. However, it imposes strict requirements on *support images*. Therefore, although few-shot paradigm algorithms can cope with the monthly updated object categories in factories, they are not yet mature for industrial implementation.

3) Benchmarking for Class-Agnostic Counting Methods: Class-agnostic counters commonly employ the few-shot paradigm, necessitating the manual selection of high-quality *support images* for accurate counting, as described in the previous section. An alternative approach involves detecting and categorizing objects of interest in the open-world through the generation of descriptive prompts. However, in industrial manufacturing settings, multiple workpieces often share similar appearances despite belonging to different categories. This presents a significant challenge when attempting to describe them solely based on brief descriptions or prompt words. Furthermore, the inclusion of human intervention in these paradigms hinders the realization of a complete end-to-end process during the testing phase. The proposed class-agnostic localization methods, in contrast to others, determine the discrimination of workpiece categories during the second stage rather than relying on manual intervention at the initial stage for category prompting. As a result, the inference stage is completely end-to-end. Table VI presents a comparison of our proposed method with existing mainstream end-to-end class-agnostic counting methods [7], [35], [36], [37], [38], [39], [40], [41] on the WPCD dataset. CAN [37], CSRNet [38], and STEERER [41] all yield remarkable results. However, these mainstream class-agnostic methods fail to generate positional information for each instance because they treat points solely as supervisory signals. Consequently, their ability to distinguish subsequent categories is limited. Furthermore, these class-agnostic regression techniques, such as [35], [36], and [39], often lead to a significant number of false points due to the existence of inconspicuous Gaussian spots and substantial overlap in dense regions, making it difficult for such methods to accurately count the number of workpieces. Also, we conducted a comparative analysis of W^E performance based on location detectors. As mentioned earlier, these location-based methods can be considered as class-agnostic detection since we only assign foreground and background labels. Our method performs well across different levels from Level-1–Level-K, with corresponding errors of 11.2, 97.7, 49.8, and 49.3. However, it does not demonstrate obvious advantages in terms of inference speed. In practical applications, it is not necessarily about pursuing faster speed, but rather striving for higher accuracy. Therefore, our method also performs exceptionally well in practical applications, as shown in Fig. 5.

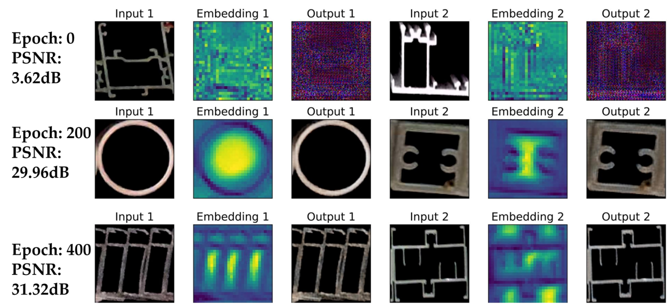

Fig. 6. Training results of WCAE at epochs 0, 200, and 400, respectively, displaying only the first two samples in each batch.

D. Pretraining for WCAE

To demonstrate the capability of WCAE in extracting salient features in embedding space, Fig. 6 visualizes the training results of WCAE at epochs 0, 200, and 400, respectively. The figure shows the first two samples in each batch, and the final loss was reduced to 0.00074 with a peak signal-to-noise ratio of 31.32 dB. The *Input* represents normalized input images, the *Embedding* represents the output features of the encoder, and the *Output* represents the reconstructed images generated by the decoder. As shown in the figure, with the progression of training, the encoder representation increasingly focuses on the unique features of the workpiece, resulting in a steady improvement in the quality of the reconstructed images. This is crucial for downstream unsupervised feature clustering tasks when K value is unknown.

E. Cluster Performance Evaluation

The detailed information of the clustered dataset is presented in Fig. 3(c), and all experiments in this section were conducted on it.

1) Comparing With Classical Methods: First, we investigated the sensitivity of the K value under the classic parameter method (KMeans++ and GMM) with different scales, dimensions, and feature extraction methods. The feature extraction methods include simple resize, ratio resize (maintaining aspect ratio, the same below), local binary pattern (LBP), ratio LBP, Gabor filtering, and ratio Gabor filtering. The dimensionality reduction methods include principal component analysis (PCA), UMAP, and t-distributed stochastic neighbor embedding (TSNE). In our experiments, the scale space was set to [32, 64, 96, 128, 160, 192], and the dimensionality space was set to [3,

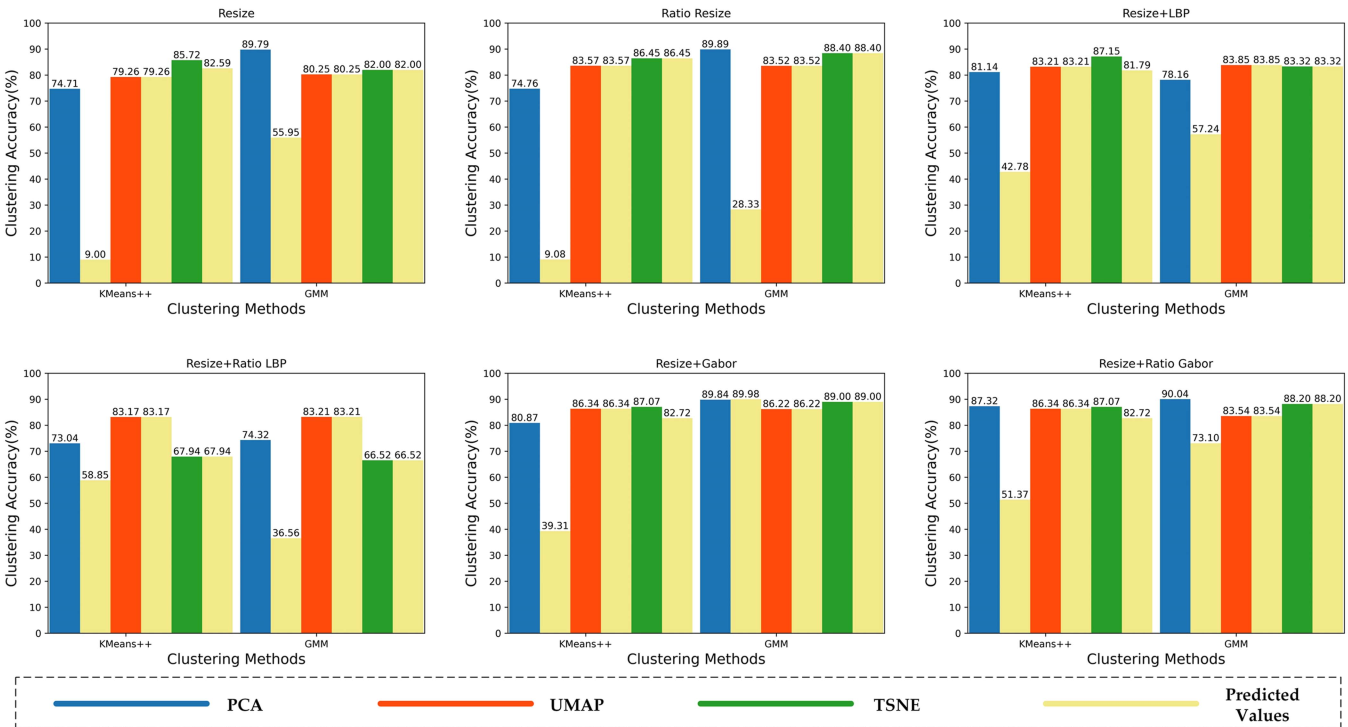

Fig. 7. Performance of various clustering methods with different feature decompositions (optimal) under a given GT K. Specifically, the clustering results obtained from PCA, UMAP, and TSNE are represented by blue, orange, and green histograms, respectively. In addition, the corresponding predicted clustering accuracies are shown in accompanying creamy–yellow histograms.

5, 16, 32, 64, 96, 128, 160, 192]. In practical settings, a counting region can hold a maximum of eight distinct workpieces (although most areas typically store only one to three types). As such, we explored the scores of CH and SC on the interval [2, 8] to determine the optimal K for qualitative analysis, *given them an unfair advantage*. Fig. 7 presents the results of different clustering methods under different feature decompositions (optimal) *given the GT* K. When their accuracies are the same, it indicates that the predicted K is the same as the GT K. From the perspective of ARI and NMI metrics, the clustering performance of GMM is superior, achieving the highest ARI and NMI scores of 85.10% and 91.43%, respectively. In addition, through experiments with traversing parameters, the accuracy of inferring the K value reached 100% when using UMAP in a 5-dim space.

2) Comparing With Nonparametric Methods: As there are only very few deep nonparametric methods available, so we have only compared DBSCAN [42], DCEC [43], DCC [44], and DeepDPM [20], which are unsupervised nonparametric methods. Prior to that, we compared the proposed methods' results under different pretraining models, including SimCLR [45], MoCov2 [46], and VQ-VAE [47]. It is evident that WCAE dominates all the metrics, achieving the highest performance (ARI: 93.34%, NMI: 95.15%, Acc: 93.32%) as given in Table VII. Similarly, it can be observed that UDCS's inferred K is closest to the GT K (Inferred K: 8). In the subsequent experiments, it is considered the best model for comparison with other methods. Fig. 7 and Table VIII demonstrate the superior performance of UDCS over current classical parametric and nonparametric methods, even in the absence of a GT K range for unfavorable comparisons. From an interpretable perspective, Fig. 8 presents the 2-D visualization of UDCS's decision on clustering boundaries in a 5-D space (with some deviation due

TABLE VII
CLUSTERING RESULTS OF UDCS UNDER VARIOUS PRETRAINED MODELS AND THEIR INFERRED K VALUES ARE PRESENTED

Pretraining method	Decomposition (dim)	ARI(%)	NMI(%)	Acc(%)	Inferred K
SimCLR [46]	TSNE(3)	50.69	66.65	59.12	8
	PCA(64)	10.11	4.26	28.90	2
	UMAP(5)	37.52	51.41	52.90	4
MoCov2 [46]	TSNE(3)	56.21	68.18	63.39	8
	PCA(64)	61.33	73.87	63.13	16
	UMAP(5)	50.55	60.07	61.68	6
VQ-VAE [47]	TSNE(3)	43.57	54.51	57.02	5
	PCA(64)	78.28	72.57	78.54	8
	UMAP(5)	38.43	61.04	42.68	7
WCAE(Ours)	TSNE(3)	**79.48**	**76.71**	**79.76**	**8**
	PCA(64)	**83.22**	**74.68**	**81.65**	**8**
	UMAP(5)	**93.34**	**95.15**	**93.32**	**8**

The bold values represent state-of-the-art performance.

TABLE VIII
COMPARING UDCS WITH THE CURRENT MAINSTREAM NONPARAMETRIC CLUSTERING METHODS

Method	ARI(%)	NMI(%)	Acc(%)	Inferred K
DBSCAN [42]	75.58	83.17	73.11	19
DCEC [43]	70.04	77.56	75.41	9
DCC [44]	79.44	83.21	80.01	7
DeepDPM [20]	88.21	92.84	87.63	8
UDCS(Ours)	**93.34**	**95.15**	**93.32**	**8**

The bold values represent state-of-the-art performance.

TABLE IX
TEST RESULTS OF TS-WPCNET BY WORKERS IN A REAL INDUSTRIAL ENVIRONMENT ARE PRESENTED

Image ID	GT number	W^E	Pre / GT K	Acc(%)	Unseen class	Image ID	GT Number	W^E	Pre / GT K	Acc(%)	Unseen class
0	154	0	2 / 2	100.00	✗	1	81	5	2 / 2	93.83	✓
2	141	0	2 / 2	100.00	✗	3	89	0	3 / 3	97.75	✓
4	384	3	1 / 1	99.22	✗	5	153	1	2 / 3	87.58	✓
6	411	4	1 / 1	99.03	✗	7	144	0	4 / 3	83.33	✓
8	446	8	2 / 2	94.62	✗	9	47	4	2 / 2	91.49	✓
10	159	1	2 / 2	99.03	✗	11	65	5	1 / 1	92.31	✗
12	69	2	2 / 2	89.71	✗	13	62	1	2 / 2	96.77	✗
14	100	0	1 / 1	100.00	✗	15	496	3	1 / 1	99.40	✗
Average Workpiece Counting Error W^E : 2.31 , Average Clustering Accuracy Acc : 95.25%											

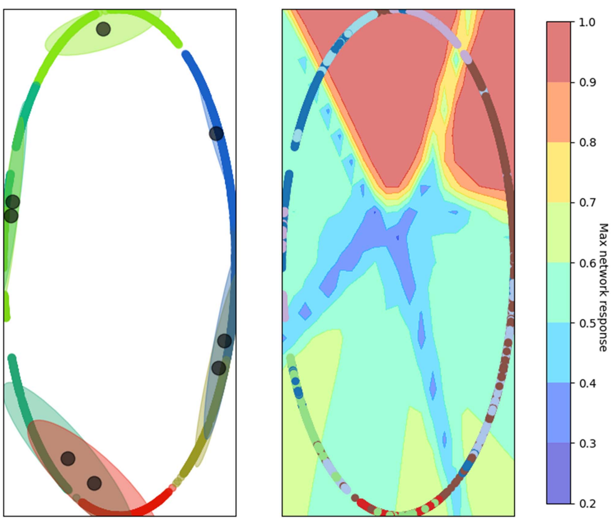

Fig. 8. Our method visualizes 2-D spatial clustering. (left-hand side) Net centers and covariances are shown, with clusters colored according to the net's assignments. (right-hand side) The decision boundary is displayed, with clusters colored according to the GT labels.

to dimensionality reduction). The left-hand side image illustrates the clustering centers assigned by the network, with each cluster colored based on the network's allocation. On the right-hand side, the image depicts the decision boundaries predicted by the network and colored according to the GT labels. Notably, it is challenging to find other unsupervised deep nonparametric methods, let alone ones with runnable code or even extending to our datasets.

F. TS-WPCNet Inference Performance Analysis

Based on the typical workflow, workers were instructed to randomly select 16 regions of interest for statistical analysis, each identified by an image identity (ID). The counting results of these regions, including *seen class* and *unseen class*, are shown in Fig. 5 (bottom). For each image, the Table IX records the GT number of workpieces, the workpiece counting error W^E, the predicted K value, the GT K value, and the clustering accuracy Acc. The last column indicates whether the image contains *unseen class*.

The evaluation indicates that TS-WPCNet achieved high localization accuracy for most images, and the average clustering accuracy reached 95.25%. The proposed method received consistent praise from workers. Nevertheless, the algorithm still has limitations, as it seems to make inference errors in K value when facing *unseen classes*, which leads to a significant decrease in clustering accuracy. There is another major reason behind the high workpiece counting error and low clustering accuracy, which may be due to a certain deviation in the output of the first-stage detector's coordinate information for each workpiece. For example, Level-2 regions are densely packed and have varying layouts, which increases the accuracy of the detector in locating the workpieces. In Level-3 scenarios, workpieces are stacked on top of each other, and although the detector may resolve each subpart of the workpieces, the rectangular boxes output by the detector can crop and capture multiple workpieces that are stacked together, resulting in a single instance containing one or more workpiece instances of the same class. Furthermore, due to the limitations of the visual sensor, the detector may fail to detect workpieces that are completely obstructed. As a result, in future work, further exploration is needed to optimize the positioning of the detector in order to efficiently and accurately accomplish the detection and counting of workpieces.

VI. CONCLUSION

This work is an attempt to lift workpiece detection and counting in industrial manufacturing into the era of foundation models. Whether TS-WPCNet attains the status of a fundamental model is yet to be determined by the extent of its adoption and application within the research community. Nonetheless, by releasing this benchmark, we believe that the WPCD dataset will provide a wide range of research opportunities for this unexplored but important area.

Going forward, we aim at investigating more precise localization methods, such as rotation-aware object detection, which predicts the rotation angle of each bounding box to avoid instance overlaps and improve clustering. Furthermore, we will explore data augmentation techniques based on generative adversarial networks to address the challenge of limited training samples. Finally, as we collect more data, we aim to extend the WPCD benchmark and evaluate our methods on larger and more diverse datasets.

REFERENCES

[1] Y. LeCun et al., "Deep learning," *Nature*, vol. 521, no. 7553, pp. 436–444, 2015.

[2] D. Tan et al., "A circular target feature detection framework based on DCNN for industrial applications," *IEEE Trans. Ind. Inf.*, vol. 17, no. 5, pp. 3303–3313, May 2021.

[3] X. Mao, X. Li, Y. Huang, J. Shi, and Y. Zhang, "Programmable logic controllers past linear temporal logic for monitoring applications in industrial control systems," *IEEE Trans. Ind. Inf.*, vol. 18, no. 7, pp. 4393–4405, Jul. 2022.

[4] X. Wang, B. Li, H. Ma, and M. Luo, "A fast quantity and position detection method based on monocular vision for a workpieces counting and sorting system," in *Proc. IEEE Chin. Control Conf.*, 2019, pp. 7906–7911.

[5] A. Aboulian et al., "NILM dashboard: A power system monitor for electromechanical equipment diagnostics," *IEEE Trans. Ind. Inf.*, vol. 15, no. 3, pp. 1405–1414, Mar. 2018.

[6] Z. Jiang, J. Liao, W. Wang, and Y. Zhai, "WPCNet: Highly accurate workpiece counting network in unrestricted scenes," in *Proc. IEEE Conf. Appl. Mach. Learn.*, 2022, pp. 255–259.

[7] D. Liang, W. Xu, Y. Zhu, and Y. Zhou, "Focal inverse distance transform maps for crowd localization," *IEEE Trans. Multimedia*, vol. 25, no. 1, pp. 6040–6052, Sep. 2022.

[8] H. Li, S. Zhang, and W. Kong, "RGB-D crowd counting with cross-modal cycle-attention fusion and fine-coarse supervision," *IEEE Trans. Ind. Inf.*, vol. 19, no. 1, pp. 306–316, Jan. 2023.

[9] M. Everingham et al., "The Pascal visual object classes (VOC) challenge," *Int. J. Comput. Vis.*, vol. 88, no. 2, pp. 303–338, 2010.

[10] T. Y. Lin et al., "Microsoft COCO: Common objects in context," in *Proc. Eur. Conf. Comput. Vis.*, 2014, pp. 740–755.

[11] M. R. Hsieh, Y. L. Lin, and W. H. Hsu, "Drone-based object counting by spatially regularized regional proposal network," in *Proc. IEEE Int. Conf. Comput. Vis.*, 2017, pp. 4165–4173.

[12] A. Osokin, D. Sumin, and V. Lomakin, "Os2D: One-stage one-shot object detection by matching anchor features," in *Proc. Eur. Conf. Comput. Vis.*, Glasgow, U.K., 2020, pp. 635–652.

[13] C. Arteta, V. Lempitsky, and A. Zisserman, "Counting in the wild," in *Proc. Eur. Conf. Comput. Vis.*, 2016, pp. 483–498.

[14] K. Song and Y. Yan, "A noise robust method based on completed local binary patterns for hot-rolled steel strip surface defects," *Appl. Surf. Sci.*, vol. 285, no. 21, pp. 858–864, 2013.

[15] P. Bergmann, M. Fauser, D. Sattlegger, and C. Steger, "MVTec AD: A comprehensive real-world dataset for unsupervise danomaly detection," in *Proc. IEEE Conf. Comput. Vis. Pattern Recognit.*, 2019, pp. 9584–9592.

[16] T. Y. Lin, P. Dollár, R. Girshick, K. He, B. Hariharan, and S. Belongie, "Feature pyramid networks for object detection," in *Proc. IEEE Conf. Comput. Vis. Pattern Recognit.*, 2017, pp. 2117–2125.

[17] Z. Cai and N. Vasconcelos, "Cascade R-CNN: High quality object detection and instance segmentation," *IEEE Trans. Pattern Anal. Mach. Intell.*, vol. 43, no. 5, pp. 1483–1498, May 2021.

[18] S. Woo et al., "CBAM: Convolutional block attention module," in *Proc. Eur. Conf. Comput. Vis.*, 2018, pp. 3–19.

[19] X. Zhou, V. Koltun, and P. Krähenbühl, "Probabilistic two-stage detection," *IEEE Trans. Geosci. Remote Sens.*, vol. 62, no. 1, 2024, Art. no. 5602214.

[20] M. Ronen, S. E. Finder, and Freifeld O. Freifeld, "DeepDPM: Deep clustering with an unknown number of clusters," in *Proc. IEEE Conf. Comput. Vis. Pattern Recognit.*, 2022, pp. 9861–9870.

[21] J. Chang and J. W. Fisher, "Parallel sampling of DP mixture models using sub-cluster splits," in *Proc. Adv. Neural Inf. Process. Syst.*, 2013, pp. 26–35.

[22] Z. Ge et al., "YOLOX: Exceeding YOLO series in 2021," *Comput. Electron. Agric.*, vol. 198, no. 1, pp. 1–13, 2022.

[23] C. Y. Wang, I. H. Yeh, and H. Y. Liao M., "You only learn one representation: Unified network for multiple tasks," *J. Inf. Sci. Eng.*, vol. 39, no. 3, pp. 691–709, 2023.

[24] S. Li, J. Yu, and H. Wang, "Damages detection of aeroengine blades via deep learning algorithms," *IEEE Trans. Instrum. Meas.*, vol. 72, no. 1, Feb. 2023, Art. no. 5009111.

[25] R. Girshick et al., "Faster R-CNN: Towards real-time object detection with region proposal networks," in *Proc. IEEE Int. Conf. Comput. Vis.*, 2015, pp. 1440–1448.

[26] K. He, G. Gkioxari, P. Dollár, and R. Girshick, "Mask R-CNN," in *Proc. IEEE Int. Conf. Comput. Vis.*, 2017, pp. 2980–2988.

[27] X. Wang et al., "SOLOv2: Dynamic and fast instance segmentation," in *Proc. Adv. Neural Inf. Process. Syst.*, 2020, pp. 17721–17732.

[28] H. Chen, K. Sun, Z. Tian, C. Shen, Y. Huang, and Y. Yan, "BlendMask: Top-down meets bottom-up for instance segmentation," in *Proc. IEEE Conf. Comput. Vis. Pattern Recognit.*, 2020, pp. 8570–8578.

[29] Z. Tian, C. Shen, X. Wang, and H. Chen, "BoxInst: High-performance instance segmentation with box annotations," in *Proc. IEEE/CVF Conf. Comput. Vis. Pattern Recognit.*, Virtual, 2021, pp. 5439–5448.

[30] Z. Tian et al., "Conditional convolutions for instance segmentation," in *Proc. Eur. Conf. Comput. Vis.*, 2020, pp. 282–298.

[31] N. Carion et al., "End-to-end object detection with transformers," in *Proc. Eur. Conf. Comput. Vis.*, 2020, pp. 213–229.

[32] Y. Li et al., "Exploring plain vision transformer backbones for object detection," in *Proc. Eur. Conf. Comput. Vis*., 2022, pp. 280–296.

[33] Z. Yang, C. Zhang, R. Li, Y. Xu, and G. Lin, "Efficient few-shot object detection via knowledge inheritance," *IEEE Trans. Image Process.*, vol. 32, no. 1, pp. 321–334, Dec. 2023.

[34] B. Sun, B. Li, S. Cai, Y. Yuan, and C. Zhang, "FSCE: Few-shot object detection via contrastive proposal encoding," in *Proc. IEEE Conf. Comput. Vis. Pattern Recognit.*, Virtual, 2021, pp. 7348–7358.

[35] M. Shi, H. Lu, C. Feng, C. Liu, and Z. Cao, "Represent, compare, and learn: A similarity-aware framework for class-agnostic counting," in *Proc. IEEE/CVF Conf. Comput. Vis. Pattern Recognit.*, 2022, pp. 9519–9528.

[36] V. Ranjan, U. Sharma, T. Nguyen, and M. Hoai, "Learning to count everything," in *Proc. IEEE Conf. Comput. Vis. Pattern Recognit.*, Virtual, 2021, pp. 3393–3402.

[37] W. Liu, M. Salzmann, and P. Fua, "Context-aware crowd counting," in *Proc. IEEE Conf. Comput. Vis. Pattern Recognit.*, 2019, pp. 5094–5103.

[38] Y. Li, X. Zhang, and D. Chen, "CSRNet: Dilated convolutional neural networks for understanding the highly congested scenes," in *Proc. IEEE Conf. Comput. Vis. Pattern Recognit.*, 2018, pp. 1091–1100.

[39] S. Abousamra et al., "Localization in the crowd with topological constraints," in *Proc. AAAI Conf. Artif. Intell.*, Virtual, 2021, pp. 872–881.

[40] B. Wang et al., "Distribution Matching for Crowd Counting," in *Proc. Adv. Neural Inf. Process. Syst.*, Virtual, Dec. 2020, pp. 1595–1607.

[41] T. Han, L. Bai, L. Liu, and W. Ouyang, "STEERER: Resolving scale variations for counting and localization via selective inheritance learning," in *Proc. IEEE/CVF Int. Conf. Comput. Vis.*, 2023, pp. 21848–21859.

[42] M. Ester et al., "A density-based algorithm for discovering clusters in large spatial databases with noise," in *Proc. 2nd Int. Conf. Knowl. Discov. Data Mining*, 1996, pp. 226–231.

[43] X. Guo et al., "Deep clustering with convolutional autoencoders," in *Proc. Int. Conf. Neural Inf. Process.*, pp. 373–382.

[44] S. A. Shah and V. Koltun, "Deep continuous clustering," in *Proc. Nat. Academy Sci.*, vol. 114, no. 37, 2017, pp. 9814–9819.

[45] T. Chen et al., "A simple framework for contrastive learning of visual representations," in *Proc. Int. Conf. Mach. Learn.*, Virtual, 2020, pp. 1597–1607.

[46] X. Chen et al., "Improved baselines with momentum contrastive learning," *Adv. Neural Inf. Process. Syst.*, vol. 34, no. 1, pp. 12251–12264, 2021.

[47] J. Peng, D. Liu, S. Xu, and H. Li, "Generating diverse structure for image inpainting with hierarchical VQ-VAE," in *Proc. IEEE/CVF Conf. Comput. Vis. Pattern Recognit.*, Virtual, 2021, pp. 10770–10779.

Ziyi Jiang (Student Member, IEEE) received the B.S. degree in communication engineering and the M.S. degree in information and communication engineering from Jiangmen Wuyi University, Jiangmen, China, in 2019 and 2023, respectively, and is currently working toward the Ph.D. degree in on radar perception and near-human space detection science with the School of Electronics and Information, South China University of Technology, Guangzhou, China.

His research interests include object perception, image processing, and pattern recognition.

Yikui Zhai (Senior Member, IEEE) received the bachelor's degree in optical electronics information and communication engineering and the master's degree in signal and information processing from Shantou University, Shantou, China, in 2004 and 2007, respectively, and the Ph.D. degree in signal and information processing from Beihang University, Beijing, China, in 2013.

Since 2007, he has been with the School of Electronics and Information Engineering, Wuyi University, Jiangmen, China, where he is currently a Full Professor and Associate Dean. He has been a Visiting Scholar with the Department of Computer Science, University of Milan, Milan, Italy, from 2016 to 2017, and from 2023 to 2024. His research interests include image processing, deep learning, and pattern recognition.

Feng Ke (Member, IEEE) received the B.S., M.S., and Ph.D. degrees in electronic engineering from the South China University of Technology (SCUT), Guangzhou, China, in 1998, 2001, and 2004, respectively.

From 2013 to 2014, he was a Visiting Scholar with the Department of Electronic Engineering, Missouri University of Science and Technology, Rolla, MO, USA. He is currently with the School of Electronic and Information Engineering, SCUT, and also with the Pazhou Laboratory, Guangzhou. He is the Vice Director with the Intelligent Communication Network and Computer Engineering Technology Engineering Center, SCUT, and the Chief Scientist with Guangdong Chinese Network Electrical Engineering Research Institute, Guangzhou, and also an Expert Advisor with Guangzhou Science and Technology Bureau, Guangzhou, and Guangdong Science and Technology Department, Guangzhou. His research interests include deep learning, resource allocation, signal processing.

Jianhong Zhou received the B.S. degree in communication engineering in 2020 from Wuyi University, Jiangmen, China, where he is currently working toward the master's in on industrial visual detection and drone vehicle detection degree with the School of Electronics and Information Engineering.

His research interests include computer vision, representational contrastive learning, image processing, and object detection.

Angelo Genovese (Senior Member, IEEE) received the Ph.D. degree in computer science from Università degli Studi di Milano, Milan, Italy, in 2014.

Since 2014, he has been a Postdoctoral Research Fellow in computer science with Università degli Studi di Milano. He has been a Visiting Researcher with the University of Toronto, Toronto, ON, Canada. Original results have been authored or coauthored in more than 30 articles in international journals, proceedings of international conferences, books, book chapters, and patents. His current research interests include signal and image processing, 3-D reconstruction, computational intelligence technologies for biometric systems, industrial and environmental monitoring systems, and design methodologies and algorithms for self-adapting systems.

Dr. Genovese is an Associate Editor of *Journal of Ambient Intelligence and Humanized Computing* (Springer).

Vincenzo Piuri (Fellow, IEEE) received the M.S. and Ph.D. degrees in computer engineering from Politecnico di Milan, Milan, Italy, in 1984 and 1988, respectively.

From 2007 to 2012, he was the Department Chair with the University of Milan, Milan, where, since 2000, he has been a Full Professor. He was an Associate Professor with Politecnico di Milan from 1992 to 2000, a Visiting Professor with The University of Texas at Austin, Austin, TX, USA, from 1996 to 1999, and a Visiting Researcher with George Mason University, Fairfax, VA, USA, from 2012 to 2016. He founded a startup company, SENSURE SRL, Bergamo, Italy. He was active in industrial research projects with several companies. His research interests include intelligent systems for industrial applications (leading it from 2007 to 2010).

Dr. Piuri is an ACM Fellow.

Fabio Scotti (Senior Member, IEEE) received the Ph.D. degree in computer engineering from Politecnico di Milano, Milan, Italy, in 2003.

Since 2015, he has been an Associate Professor of computer science with Università degli Studi di Milano, Milan. Original results have been authored or authored in more than 100 papers in international journals, proceedings of international conferences, books, book chapters, and patents. His current research interests include biometric systems, machine learning and computational intelligence, signal and image processing, theory and applications of neural networks, 3-D reconstruction, industrial applications, intelligent measurement systems, and high-level system design.

Dr. Scotti is an Associate Editor for IEEE TRANSACTIONS ON HUMAN MACHINE SYSTEMS and *Soft Computing* (Springer).