Point Pattern Feature-Based Anomaly Detection for Manufacturing Defects, in the Random Finite Set Framework

AMMAR MANSOOR KAMOONA1,2, (student member,IEEE), AMIRALI KHODADADIAN GOSTAR 1, ALIREZA BAB-HADIASHAR AND REZA HOSEINNEZHAD.1,

1School of Engineering, RMIT University, Melbourne, Victoria, Australia
2Department of Electrical Engineering, Faculty of Engineering, University of Kufa, Najaf, Iraq

Corresponding author: Ammar Kamoona (e-mail: s3442819@student.rmit.edu.au).

ABSTRACT Defect detection in the manufacturing industry is of utmost importance for product quality inspection. Recently, optical defect detection has been investigated as anomaly detection using different deep learning methods. Most current works employ feature extraction methods that describe the entire image using a single feature vector, called the global feature. However, the use of the global feature is affected by changes in several factors, such as lighting and viewpoint changes. An alternative is to use point pattern features known as local features or keypoints which are robust to changes in conditions mentioned earlier. The use of robust point pattern features, such as SIFT, for defect detection using the recently developed set-based methods is not yet explored. This paper proposes the use of point pattern features within a random finite set framework for defect detection. Also, we evaluate different point pattern feature detectors and descriptors, handcrafted point pattern features (e.g., SIFT), and pre-trained deep features, for defect detection applications. Experiments on a large-scale defect detection dataset (MVTec-AD) are carried out. The results are compared with state-of-the-art global feature-based anomaly detection methods. Results show that using point pattern features as data points within random finite set-based anomaly detection, achieves the most consistent defect detection accuracy on the MVTec-AD dataset. Also, this evaluation shows that transfer learning of deep features has promising results for defect detection.

INDEX TERMS Anomaly detection, defect detection, random finite set, point pattern features, local features, transfer learning.

I. INTRODUCTION

AUTOMATED visual inspection is a common part of the quality control process in many modern manufacturing applications. The main aim is to rectify limitations of human visual inspection, such as manual visual inspection errors, including missing or incorrect identification of defects. Such errors can significantly impact product quality, lead to unnecessary production costs and increase overall waste. In light of this, an automated computer vision-based visual inspection can significantly improve productivity [1]. Different data-driven methods based on deep learning have been proposed for visual inspection in different application areas, such as manufacturing [2]–[5], construction [6]–[8], transportation [9], [10] and computing systems [11], [12].

A common approach to develop automated defect detection is to employ various image processing algorithms. Some examples include Haar filter for tile surface inspection [13], local order binary pattern for fabric defect detection [14], and scale-invariant keypoint features for PCB inspection [15]. Furthermore, a combination of image processing and machine learning algorithms has shown satisfactory performance. An example is to use the histogram of oriented gradients (HOG) features with a support vector machine (SVM) classifier for railway [16] and Fabric [17] inspections.

A significant drawback of the solutions that use traditional image processing techniques is the need for implicit engineered features, which can be challenging when applied to complex cases. An attempt to address this issue has been to employ deep learning-based solutions for automated defect detection. The advancement of deep learning methods in computer vision [18], [19] has paved the way to develop reliable visual inspection systems. These techniques use data
representation learning to perform different tasks, where the goal is to transform complex data into abstract representations known as features. Wang et al. [4] highlighted the power of deep learning and its potential opportunities for smart manufacturing. Various deep learning techniques have been proposed to deal with different surface defects [20], [21]. Furthermore, Hui et al. [22] proposed a LEDNet network for defect detection and classification on LED chips. Cha et al. [6] proposed structural damage detection using Faster R-CNN to detect five different defects.

The performance of deep learning methods, such as convolutional neural networks (CNNs), are limited by the training samples’ availability, giving rise to two problems: the class imbalance between the normal and defected samples and difficulty of data annotation. Although collecting many unlabeled images (usually normal samples without defects) could be a simple task, labeling these samples is expensive and require a trained inspector, and often human error can occur. These two problems are well-known and subject to continuing research [23]–[25].

The limitations introduced by class imbalance and lack of a vast number of annotated samples for training can be addressed by treating the defect detection problem as an anomaly detection problem in which the detection decision is made based on deviation from the statistical distribution of “normal” samples. In addition, there is no difference between the definition of defect detection stated by Newman et al. [26] and anomaly detection.

From this perspective, Bergmann et al. [27] proposed evaluating different deep anomaly detection methods for defect detection. They also presented a new dataset (MVTec-AD), including various objects and textures with different defects. Existing works [27], [28] in this domain mainly concentrate on performing anomaly detection by looking at the whole image through extracting a single feature vector and its deviation from the “normal” samples, summarizing the entire image by one feature vector commonly referred to in the literature by “global feature” [29]. The global feature is effective at describing the content of an entire image, while local features are suitable for describing the geometrical information about a specific part of the image. In defect detection, detecting the slight variation at a specific part of the image is of interest, (see Figure 3). This implies that using local features is more intuitive to capture the slight variation.

To the best of our knowledge, no work in the literature has explored the use of local features or point pattern features for defect detection. This paper explores the use of different point pattern features for defect detection and shows how they can be directly analyzed via a multiple instance learning routine designed in the Random Finite Set (RFS) framework.

Vo et al. [30] proposed to model the point pattern features (e.g., SIFT) using point processes and use multiple instances learning for anomaly detection. The proposed approach is based on modeling each point pattern feature as an RFS sample and includes the introduction of a likelihood function based on the RFS density assumptions (e.g. Poisson). The performance of deep learning methods, such as convolutional neural networks (CNNs), are limited by the training samples’ availability, giving rise to two problems: the class imbalance between the normal and defected samples and difficulty of data annotation. Although collecting many unlabeled images (usually normal samples without defects) could be a simple task, labeling these samples is expensive and require a trained inspector, and often human error can occur. These two problems are well-known and subject to continuing research [23]–[25].

The limitations introduced by class imbalance and lack of a vast number of annotated samples for training can be addressed by treating the defect detection problem as an anomaly detection problem in which the detection decision is made based on deviation from the statistical distribution of “normal” samples. In addition, there is no difference between the definition of defect detection stated by Newman et al. [26] and anomaly detection.

From this perspective, Bergmann et al. [27] proposed evaluating different deep anomaly detection methods for defect detection. They also presented a new dataset (MVTec-AD), including various objects and textures with different defects. Existing works [27], [28] in this domain mainly concentrate on performing anomaly detection by looking at the whole image through extracting a single feature vector and its deviation from the “normal” samples, summarizing the entire image by one feature vector commonly referred to in the literature by “global feature” [29]. The global feature is effective at describing the content of an entire image, while local features are suitable for describing the geometrical information about a specific part of the image. In defect detection, detecting the slight variation at a specific part of the image is of interest, (see Figure 3). This implies that using local features is more intuitive to capture the slight variation.

To the best of our knowledge, no work in the literature has explored the use of local features or point pattern features for defect detection. This paper explores the use of different point pattern features for defect detection and shows how they can be directly analyzed via a multiple instance learning routine designed in the Random Finite Set (RFS) framework.

Vo et al. [30] proposed to model the point pattern features (e.g., SIFT) using point processes and use multiple instances learning for anomaly detection. The proposed approach is based on modeling each point pattern feature as an RFS sample and includes the introduction of a likelihood function based on the RFS density assumptions (e.g. Poisson).
other approach uses transfer learning (pretrained network) for anomaly detection [37], [38]. These networks use the feature of pretrained networks to detect anomalies, and they are based on the assumption that the feature space of these networks is generic enough to be mapped to dissimilar tasks [39]. Andrews et al. [37] use VGG features with a One-Class Support Vector Machine (OCSVM) for anomaly detection. Most recently, Rippel et al. [40] proposed to use Efficient Net [41] features, and model these features using multivariate Gaussian distribution. Anomaly detection as a one-class classification problem is also being considered. One major family is based on predicting geometric transformation [42], [43], such as rotation, translation. Other family is based on contrastive learning with geometric augmentations [44], [45]. These approaches succeed only for semantic anomaly detection benchmark such as ImageNet [46]. Our work differs from previous works by the using transfer learning of deep local feature instead of commonly used global feature for defect detection to describe geometrical information of image regions. These features provide an efficient memory representation. In addition, these features are proven to be robust against noise [47] and illumination changes.

III. RANDOM FINITE SET-BASED ANOMALY DETECTION

The random finite set-based anomaly detection method models the point pattern features as an RFS. The rationale is that in a feature set

\[ X = \{x_1, x_2, \ldots, x_n\}, \]

both the feature elements \(x_i\) (\(i = 1, \ldots, n\)) and the number of features (also called its “cardinality”), \(|X| = n\), vary randomly. Accordingly, each measured set of point features \(X = \{x_i\}_{i=1}^{n}\) is treated as an RFS. The RFS density with respect to some measure \(U\) is given by [48]:

\[ p(X) = p_c(|X|)(|X|!U^{|X|}p(x_1, \ldots, x|X|)), \]

where \(p_c(n) = \Pr(|X| = n)\) is the cardinality distribution, \(U\) is the unit of hyper-volume, and \(p(x_1, \ldots, x_n)\) is a symmetric joint feature density for the given cardinality \(|X| = n\) [49].

Different assumptions about the mathematical form of the RFS density \(p(X)\) can be considered in practice. Examples include Poisson RFS [50], Beta RFS [51], the Bernoulli RFS [48], the multi-Bernoulli RFS [52], the IDD-cluster RFS [53], the labeled multi-Bernoulli RFS [54] and finally the generalized labeled multi-Bernoulli RFS [55]. The densities, as mentioned earlier, have been used for multi-object tracking in which they treat the multi-object entity as a random finite set. An independent identical distributed (IID) cluster RFS density is given as follows:

\[ p(X) = p_c(|X|)(|X|!U^{|X|}p(\cdot)^{|X|}), \]

where \(p(\cdot)\) is the feature density and \(p(\cdot)^{|X|} = \prod_{x \in X} p(x)\) is the finite set exponential. Assuming that the cardinality distribution follows Poisson distribution, the IID-cluster RFS turns into Poisson RFS given by:

\[ p(X) = \rho^{|X|} \exp(-\rho) |U p(\cdot)|^X, \]

where \(\rho\) is the non-negative Poisson intensity. Equation (3) can be rewritten using finite set exponential as follows:

\[ p(X) = \rho^{|X|} \exp(-\rho) |U|^X \prod_{x \in X} p(x). \]

The set density, \(\prod_{x \in X} p(x)\), in equation (4) is strongly dependent on cardinality and returns relatively large values for small numbers of keypoint samples. Therefore, such an RFS density can return poor results when used for anomaly detection. Vo et al. [30] propose to use the following “ranking” function to infer normality with IID cluster assumptions for set density:

\[ r(X) = p_c(|X|) \left( \frac{p(\cdot)}{||p(\cdot)||_2} \right)^{|X|}, \]

where \(||p(\cdot)||_2^2\) is the squared \(L^2\)-norm of \(p(\cdot)\). For a Poisson RFS density, the ranking function turns into:

\[ r(X) = \rho^{|X|} \exp(-\rho) \prod_{x \in X} \frac{p(x)}{|X|! ||p(x)||_2^2}. \]

IV. EVALUATION METHODOLOGY

In this section, we discuss the use of RFS-based anomaly detection for defect detection from image analysis. The general framework in which such solutions can be developed is shown in Figure 2. It comprises two distinct parts: point pattern feature extraction; and RFS-based analysis and inference for defect detection.

A. POINT PATTERN FEATURE EXTRACTOR AND DESCRIPTOR

The term point pattern feature refers to a set of keypoints (interest points or local features) – rather than a vector-based feature or global features – returned by any feature extraction pipeline method. These keypoints are usually 2D locations in an image that should be stable and repeatable against different lighting conditions and viewpoints. Point pattern features have been used in different computer vision tasks, such as visual SLAM, camera calibration, Structure-from-motion (SfM), and image matching [47]. Generally, most local feature detection methods, such as SIFT [15], return an output in the form of a set. In contrast, global feature detection methods commonly return features in vector format, such as in the Histogram of Oriented Gradients (HOG) [56]. In machine learning, the traditional approach is to convert the point pattern features into vector format using different methods such as bag of visual word [31].

One of the most well-known handcrafted point pattern feature detection methods, which is also employed in this evaluation, is a Harris-Laplace point detector that uses the Harris corner detector to detect scale-invariant keypoints. Then, around each keypoint, a descriptor such as SIFT [15]...
Due to the lack of access to the defected samples, unsupervised anomaly detection is a preferred option for defect detection [27]. In this approach, only the normal samples (defect-free) are used in the training phase. Similarly, the RFS-based defect detection only uses the normal samples during training to maximize RFS set density in which the parameters of the model are learned using either the maximum likelihood estimator (MLE) or expectation maximization (EM) [51].

1) Training phase

In this phase, the goal is to learn the random finite set likelihood function parameters that best fit the distribution of defect-free samples by maximizing their likelihood using a maximum likelihood estimator (MLE). Consider a Poisson IID RFS in which the cardinality distribution parameterized by \( \rho \) and denoted as \( p_c(|X|; \rho) \), and multi-feature joint density is parameterized by \( \theta \) and denoted as \( p(x_1, \ldots, x_{|X|}; \theta) \).

The goal of the training phase is to estimate these parameters as follows:

\[
(\hat{\rho}, \hat{\theta}) = \arg \max_{\rho, \theta} \prod_{X \in \mathcal{O}} \mathcal{L}(\rho, \theta | X),
\]

where \( \mathcal{O} \) is the ensemble of all training feature sets (each being a single set of image descriptors in the training dataset of normal samples). In practice, there is usually no prior information on the parameters and the prior density \( p(\rho, \theta) \) is uniform (constant). Hence,

\[
(\hat{\rho}, \hat{\theta}) = \arg \max_{\rho, \theta} \prod_{X \in \mathcal{O}} \mathcal{L}(X|\rho, \theta).
\]

With Poisson assumption for RFS density, equation (8) is simplified to the following:

\[
(\hat{\rho}, \hat{\theta}) = \arg \max_{\rho, \theta} \prod_{X \in \mathcal{O}} p_c(|X|; \rho) |X|! [p(\cdot; \theta)]^{|X|}. \tag{9}
\]

Vo et al. [30] proved that the above optimization can be decomposed into two separate optimizations for the cardinality parameter and feature distribution as follows:

\[
\hat{\rho} = \arg \max_{\rho} \prod_{X \in \mathcal{O}} p_c(|X|; \rho) \quad \tag{10}
\]

\[
\hat{\theta} = \arg \max_{\theta} \prod_{X \in \mathcal{O}} [p(\cdot; \theta)]^{|X|}. \tag{11}
\]

Substituting \( p_c(|X|; \rho) \) with \( \rho^{|X|} e^{-\rho} / |X|! \) in equation (10), the solution simply turns out as average of all training feature set cardinality values,

\[
\hat{\rho} = \frac{1}{|\mathcal{O}|} \sum_{X \in \mathcal{O}} |X|.
\]

Assuming that all point features within each feature set are IID vectors in \( \mathbb{R}^D \) and distributed according to a Gaussian density with parameters \( \theta = (\mu, \Sigma) \), equation (11) turns into:

\[
(\hat{\mu}, \hat{\Sigma}) = \arg \max_{(\mu, \Sigma)} \prod_{X \in \mathcal{O}} \prod_{x \in X} \mathcal{N}(x; \mu, \Sigma), \tag{13}
\]
where $\mathcal{N}(\cdot; \mu, \Sigma)$ is the multivariate Gaussian density function with mean $\mu \in \mathbb{R}^D$ and covariance $\Sigma \in \mathbb{R}^{D \times D}$. The above optimization has a closed-form solution as follows:

\[
\hat{\mu} = \sum_{x \in O} \sum_{x \in X} x / \sum_{x \in O} |X| (14)
\]

\[
\hat{\Sigma} = \sum_{x \in O} \sum_{x \in X} (x - \hat{\mu})(x - \hat{\mu})^T / \sum_{x \in O} |X|. (15)
\]

Equations (14) and (15) are used to estimate the mean and covariance of a single Gaussian component.

The single object density $p(x)$ can also be modeled as a mixture of Gaussian components. In that case, parameters of the Gaussian components can be learned using the Expectation-Maximization (EM) algorithm. EM algorithm requires to define the number of Gaussian components as prior. Choosing the correct number of components for unlabeled data is a challenging task and requires a lot of fine-tuning with a substantial effect on the performance; see the ablation study Section V-D. Therefore, in this paper, we propose to use variations inference Bayes (VB) \cite{66} method for parameter estimation.

After parameters of RFS ranking are estimated, the RFS rank log-likelihood of all training images is calculated again, and $p$-quantile of the likelihood is set to define the threshold value $L_{th}$ to be used in the testing phase.

2) Testing Phase

In this section, defect detection using a random finite set is introduced, as shown in Algorithm 1. The inputs to the algorithm are the RGB image $I$ to be inspected, RFS density parameters $\rho$, $\mu$, $\Sigma$ obtained from the training phase, and the likelihood threshold $L_{th}$. First, keypoints and their descriptors are extracted from $I$ by calling the function KEYPOINT_DETECTOR($\cdot$). This can represent any of the point pattern feature detection routines explained in Section V-B. The output is a set of keypoints $p$ and their corresponding descriptors $X$. Most descriptors, handcrafted and deep learning-based ones, are in high dimension; thus, principal component analysis (PCA) is carried out to reduce the dimension of the descriptor to two dimensions denoted by $X_{\text{red}}$. Finally, the RFS log-likelihood rank $L$ for the image set descriptors is calculated from (6) by executing the function LOG_RANK($\cdot; \rho$, $\mu$, $\Sigma$), and compared with the threshold value $L_{th}$. If the result is lower than $L_{th}$, the defect detection flag is triggered.

V. EXPERIMENTS

A. DATASET

MVTec-AD dataset: MVTec-AD \cite{27} is a comprehensive and challenging real-world industrial image dataset developed for defect detection. The dataset has an extensive collection of texture and object images. It has 5354 high-resolution color images of a variety of objects and textures. In MVTec-AD, there are 15 different categories (ten objects and five textures). Each object has only normal samples for training and normal and defect samples for testing. There are 70 different types of defects, such as scratches, dents, contamination, and various structural deformations. Figure 3 shows examples of these samples. The first row shows the defect-free samples, while the second row shows defect samples.

B. FEATURE EXTRACTION

To extract sparse local features from the defect-free samples, different point pattern feature extraction methods have been used in this evaluation. The focus of this paper is on the current state-of-the-art deep learned features. Thus, the following deep learning point pattern feature detection and descriptor have been used. LF-net \cite{63}: is an unsupervised learned network that uses the detect-then-describe strategy in one end-to-end network. The network outputs a descriptor with dimension 255-D. We use the LF-net pretrained on ScanNet \cite{67} dataset with over 2.5M images obtained from GitHub\cite{64}. D2-net \cite{64}: this network provides joint learning for detection and description trained on MegaDepth dataset \cite{68} obtained from GitHub. D2-Net keypoint detection is based on local maxima over all the CNN channels and the spatial dimensions of the feature maps. During the inference, the network provides a multi-scale option, which we call this network as (D2-net2). r2d2 \cite{65}: this network provides joint learning of detection and description. This network explicitly learns both keypoint repeatability and repeatability from the training set to avoid ambiguous areas. r2d2 network is a self-supervised trained network using synthetic and real images.

Finally, we have used the most well-known handcrafted feature detection and description, which are SIFT \cite{15} and Harris-Laplace point detector, because they have shown good

\[1\] https://github.com/vcg-uvic/lf-net-release

\[2\] https://github.com/mihaidusmanu/d2-net

\[3\] https://github.com/naver/r2d2
Algorithm 1 RFS-based defect detection from images.

1: procedure DEFECT_DETECTION(I; ρ, μ, Σ, Lth) \(\triangleright\) I : image, ρ, μ, Σ : density parameters, Lth : Likelihood threshold.
2: \((pts, X) \leftarrow KEPOINT_DETECTOR(I)\) \(\triangleright\) A set of keypoints pts and their descriptors X are extracted from the I.
3: \(X_{\text{red.}} \leftarrow \text{PCA}(X)\) \(\triangleright\) The dimension of each descriptor \(x \in X\) is reduced to 2 using PCA.
4: \(L \leftarrow \text{LOG_RANK}(X_{\text{red.}}; ρ, μ, Σ)\) \(\triangleright\) Log-likelihood is calculated from RFS rank function.
5: \(η \leftarrow 0\)
6: if \(L < L_{\text{th}}\) then \(η \leftarrow 1\)
7: end if
8: return \(η\)
9: end procedure

10: function LOG_RANK(X; ρ, μ, Σ)
11: \(σ \leftarrow 0\)
12: for \(x \in X\) do
13: \(p(x) \leftarrow \mathcal{N}(x; μ, Σ); \ σ \leftarrow σ + \log p(x)\) \(\triangleright\) Density terms are computed and \(\sum_{x \in X} \log p(x)\) is gradually formed.
14: end for
15: \(L \leftarrow |X| \log ρ - ρ - σ \log (|X|) + \frac{\log 2}{2} |X|^2\) \(\triangleright\) Log-rank is calculated from (6) where \(||p(\cdot)||_2^2\) is \(2^{-\frac{|x|}{2}}\).
16: return \(L\)
17: end function

For category recognition, for SIFT and Harris detector, we use the VLFeat\(^4\) library implementation. Figure 4 shows the keypoints of different feature extraction methods.

**FIGURE 4.** Point pattern features (keypoints) of different feature extraction methods.

C. EXPERIMENTAL RESULTS

The proposed RFS-based defect detection using different point pattern feature detection and description is evaluated on MVTec-AD dataset. For the sake of comparison with deep models, we have compared the proposed approach with the deep anomaly detection given in [27] which are as follows: AE(SSIM): deep auto-encoder using SSIM as loss function. AE(L2): deep auto-encoder using pixel-wise \(L_2\) loss function. AnoGAN: anomaly detection using Generative adversarial networks. CNN: convolutional neural network.

\(^4\)https://www.vlfeat.org/

feature dictionary [69]. **Texture inspection:** Gaussian mixture model for texture inspection [70]. **Variation model:** using variational model [71] of GMM for non-textured images is used by providing prior alignment of the object contours.

The implementation details and setups of these methods can be found in [27]. For evaluation, we have used the same evaluation metric used in [27] in reporting the defect detection accuracy. We use recall and specificity defined as fellow:

\[
\text{Recall} = \frac{TP}{TP + FN}, \quad \text{Specificity} = \frac{TN}{TN + FP},
\]

where \(TP\) is the number of the true positives, \(FN\) is the number of false negatives, \(TN\) is the number of true negatives, and \(FP\) is the number of false positives. Recall (Re.) reflects the ratio of correctly classifying defect samples, and specificity (Sp.) reflects the ratio of correctly classifying normal samples.

In practice, a good detector should always have a high true positive ratio, a high recall, and a low false alarm rate, a high specificity. Having a high false alarm means a lot of good samples have been classified as defect samples. This increases the waste in production and money and not of interest [72], [73]. From this perspective, we have recorded the mean of both recalls and specificity and reported in Table 2 and Table 3.

Table 1 shows the ratio of correctly classified (normal and defect) samples for each object and the mean of these ratios. We rank each object’s mean (lower is better) of these methods, and the final rank of the average (Avg.) rank is shown in the last row. The RFS-based defect detection using (SIFT) has the best performance (ranks first) followed by auto-encoder-based methods (rank second and third). RFS (LF-net) and RFS (r2d2) show better performance compared to AnoGan and CNN-dictionary methods. We can see that RFS-based defect detection shows a promising performance...
Table 1. Performance results of Poisson RFS rank-based defect detection using different feature extraction methods on MVTec-AD dataset. The specificity (Sp.), recall (Re), and their mean are given. The best means are in bold.

| Category  | Sp. | Re. | Mean | Rank | Sp. | Re. | Mean | Rank | Sp. | Re. | Mean | Rank | Sp. | Re. | Mean | Rank | Sp. | Re. | Mean | Rank | Sp. | Re. | Mean | Rank | Sp. | Re. | Mean | Rank |
|-----------|-----|-----|------|------|-----|-----|------|------|-----|-----|------|------|-----|-----|------|------|-----|-----|------|------|-----|-----|------|------|-----|-----|------|------|
| Zipper    | 0.75 | 0.75 | 0.59 | 0.78 | 0.78 | 0.00 | 0.00 | 0.00 | 1.00 | 0.97 | 0.78 | 0.78 | -    | 1.00 | 0.00 | 0.00 | 0.00 | 0.60 | 0.63 | 0.40 | 0.29 | -    | 0.80 | 0.80 | 0.59 | 0.54 |
| Transistor| 0.82 | 0.81 | 0.96 | 0.96 | 0.93 | 0.91 | 1.00 | 0.97 | 0.97 | 0.97 | 0.97 | 0.97 | 1.00 | 0.97 | 0.97 | 0.97 | 0.97 | 0.73 | 0.77 | 0.77 | 0.77 | 0.73 | 0.77 | 0.77 | 0.77 | 0.77 |
| Toothbrush| 0.95 | 0.95 | 0.95 | 0.95 | 0.95 | 0.95 | 0.95 | 0.95 | 0.95 | 0.95 | 0.95 | 0.95 | 0.95 | 0.95 | 0.95 | 0.95 | 0.95 | 0.95 | 0.95 | 0.95 | 0.95 | 0.95 | 0.95 | 0.95 | 0.95 | 0.95 |
| Screw     | 0.63 | 0.63 | 0.63 | 0.63 | 0.63 | 0.63 | 0.63 | 0.63 | 0.63 | 0.63 | 0.63 | 0.63 | 0.63 | 0.63 | 0.63 | 0.63 | 0.63 | 0.63 | 0.63 | 0.63 | 0.63 | 0.63 | 0.63 | 0.63 | 0.63 | 0.63 |
| Bottle    | 0.70 | 0.70 | 0.70 | 0.70 | 0.70 | 0.70 | 0.70 | 0.70 | 0.70 | 0.70 | 0.70 | 0.70 | 0.70 | 0.70 | 0.70 | 0.70 | 0.70 | 0.70 | 0.70 | 0.70 | 0.70 | 0.70 | 0.70 | 0.70 | 0.70 | 0.70 |
| Cable     | 0.89 | 0.89 | 0.89 | 0.89 | 0.89 | 0.89 | 0.89 | 0.89 | 0.89 | 0.89 | 0.89 | 0.89 | 0.89 | 0.89 | 0.89 | 0.89 | 0.89 | 0.89 | 0.89 | 0.89 | 0.89 | 0.89 | 0.89 | 0.89 | 0.89 | 0.89 |
| Capsule   | 0.78 | 0.78 | 0.78 | 0.78 | 0.78 | 0.78 | 0.78 | 0.78 | 0.78 | 0.78 | 0.78 | 0.78 | 0.78 | 0.78 | 0.78 | 0.78 | 0.78 | 0.78 | 0.78 | 0.78 | 0.78 | 0.78 | 0.78 | 0.78 | 0.78 | 0.78 |
| Hazelnut  | 0.95 | 0.95 | 0.95 | 0.95 | 0.95 | 0.95 | 0.95 | 0.95 | 0.95 | 0.95 | 0.95 | 0.95 | 0.95 | 0.95 | 0.95 | 0.95 | 0.95 | 0.95 | 0.95 | 0.95 | 0.95 | 0.95 | 0.95 | 0.95 | 0.95 | 0.95 |
| Avg.Rank  | 2.5  | 5.5  | 5.9  | 5.4  | 4.8  | 4.5  | 4.4  | 2.8  | 6.4  | 7.9  | 6.3  | 6.3  | 6.3  | 6.3  | 6.3  | 6.3  | 6.3  | 6.3  | 6.3  | 6.3  | 6.3  | 6.3  | 6.3  | 6.3  | 6.3  | 6.3  |

Table 3 shows the performance results of five texture images. Similar to Table 1, the final rank has also been recorded. The RFS (SIFT) method has the best performance (ranks the first), and CNN-dictionary has the second better performance (ranks the second) compared to others. In conclusion, we can see from Tables 1 and 2 that RFS (SIFT) has the best performance compared to the different deep learning methods. Also, RFS (LF-net) and RFS (r2d2) show better consistent performance for texture and object images.

We also included evaluation of the proposed Poisson RFS (SIFT) using the most common metrics such as precision, recall (Re), and their mean are given. The best means are in bold. The specificity (Sp.), recall (Re), and their mean are given. The best means are in bold.

D. ABLATION STUDY

In this section, we study the effect of using different parameters on RFS rank defect detection performance. We study the effect of choosing a different number of Gaussian components for modeling the single descriptor density on the performance using the mean performance, as shown in the third row of Table 1. Figures 5 and 6 show the mean performance of different objects and textures using RFS rank with different numbers of Gaussian components using SIFT keypoint descriptors. It is clear that the performance varies with the number of Gaussian components. We employ variational Bayesian inference for Gaussian Mixture Model [66] to choose the correct number of components.

This work is licensed under a Creative Commons Attribution 4.0 License. For more information, see https://creativecommons.org/licenses/by/4.0/
TABLE 2. Performance results of Poisson RFS rank-based defect detection on MVTec-AD dataset. The specificity (Sp.), recall (Re.), and their mean are given. The best means are in **bold**.

| Category  | RFS (SIFT) | RFS (Harris) | RFS (D2-net) | RFS (D2-net2) | RFS (LF-net) | RFS (r2d2) | AE (SSIM) | AE (L2) | AnoGAN | CNN | Texture inspection |
|-----------|------------|--------------|--------------|--------------|--------------|------------|-----------|----------|--------|-----|-------------------|
| Carpet    |            |              |              |              |              |            |           |          |        |     |                   |
|           | Sp. 0.71   | 0.28         | 0.96         | -            | 0.42         | 0.42       | 0.43      | 0.57    | 0.82   | 0.89 | 0.57              |
|           | Re. 0.70   | 0.83         | 0.08         | -            | 0.76         | 0.79       | 0.90      | 0.42    | 0.16   | 0.36 | 0.61              |
|           | Mean **0.70** | 0.55       | 0.52         | -            | 0.59         | 0.61       | 0.66      | 0.49    | 0.49   | 0.62 | 0.59              |
|           | Rank 1     | 6            | 7            | -            | 5            | 4          | 2         | 8       | 8      | 3   | 5                 |
| Grid      |            |              |              |              |              |            |           |          |        |     |                   |
|           | Sp. 0.52   | 0.52         | 0.66         | -            | 0.00         | 0.80       | 0.38      | 0.57    | 0.90   | 0.57 | 1.00              |
|           | Re. 0.86   | 0.52         | 0.56         | -            | 1.00         | 0.36       | 1.00      | 0.98    | 0.12   | 0.33 | 0.05              |
|           | Mean 0.69  | 0.52         | 0.61         | -            | 0.50         | 0.58       | 0.69      | **0.77** | 0.51  | 0.45 | 0.52              |
|           | Rank 2     | 5            | 3            | -            | 7            | 4          | 2         | 1       | 6     | 8   | 5                 |
| Leather   |            |              |              |              |              |            |           |          |        |     |                   |
|           | Sp. 0.93   | 0.34         | -            | -            | 0.00         | 0.34       | 0.00      | 0.06    | 0.91   | 0.63 | 0.00              |
|           | Re. 0.40   | 0.73         | -            | -            | 1.00         | 0.78       | 0.92      | 0.82    | 0.12   | 0.71 | 0.99              |
|           | Mean 0.66  | 0.54         | -            | -            | 0.50         | 0.56       | 0.46      | 0.44    | 0.51   | **0.67** | 0.49         |
|           | Rank 2     | 4            | -            | -            | 6            | 3          | 8         | 9       | 5      | 1   | 7                 |
| Wood      |            |              |              |              |              |            |           |          |        |     |                   |
|           | Sp. 1.00   | -            | -            | -            | 0.94         | 0.94       | 0.84      | 1.00    | 0.89   | 0.79 | 0.42              |
|           | Re. 0.78   | 0.31         | -            | -            | 0.56         | 0.30       | 0.82      | 0.47    | 0.47   | 0.88 | 1.00              |
|           | Mean **0.89** | 0.65       | -            | -            | 0.75         | 0.62       | 0.83      | 0.73    | 0.68   | 0.83 | 0.71              |
|           | Rank 1     | 7            | 3            | -            | 8            | 1          | 2         | 4       | 6     | 2   | 5                 |
| Tile      |            |              |              |              |              |            |           |          |        |     |                   |
|           | Sp. 0.42   | 0.24         | 1.00         | 1.00         | 0.78         | 0.33       | 1.00      | 1.00    | 0.97   | 0.97 | 1.00              |
|           | Re. 0.70   | 0.86         | 0.00         | 0.01         | 0.55         | 0.80       | 0.04      | 0.54    | 0.05   | 0.44 | 0.43              |
|           | Mean 0.56  | 0.55         | 0.50         | 0.50         | 0.67         | 0.57       | 0.52      | **0.77** | 0.51  | 0.70 | 0.71              |
|           | Rank 5     | 6            | 9            | 9            | 3            | 4          | 7         | 1       | 8     | 2   | 2                 |
| Avg. Rank | 2.2        | 5.6          | 6.3          | 9            | 4.8          | 4.6        | 4.2       | 4.6     | 6.6    | 3.2 | 4.8               |
| Rank      | **1**      | 6            | 7            | 9            | 5            | 4          | 3         | 4       | 8     | 2   | 5                 |

FIGURE 5. The mean performance of RFS rank defect detection of MVTec-AD objects for different Gaussian components with a number of iteration set to 30.

TABLE 3. Precision, Recall, F1-score, of the proposed Poisson RFS (SIFT) for MVTec-AD dataset.

| Category  | RFS (SIFT) | RFS (Harris) | RFS (D2-net) | RFS (D2-net2) | RFS (LF-net) | RFS (r2d2) | AE (SSIM) | AE (L2) | AnoGAN | CNN | Texture inspection |
|-----------|------------|--------------|--------------|--------------|--------------|------------|-----------|----------|--------|-----|-------------------|
| Zipper    | Precision 0.93 | Recall 0.89 | F1-score 0.91 | Precision 0.73 | Recall 0.55 | F1-score 0.83 | Precision 0.81 | Recall 0.73 | F1-score 0.77 | Precision 0.94 | Recall 0.61 | F1-score 0.73 | Precision 0.91 |
| Transistor| Precision 0.91 | Recall 0.45 | F1-score 0.83 | Precision 0.88 | Recall 0.67 | F1-score 0.56 | Precision 0.95 | Recall 0.45 | F1-score 0.61 | Precision 1.00 | Recall 0.67 | F1-score 0.80 | Precision 0.80 |
| Toothbrush| Precision 0.88 | Recall 0.49 | F1-score 0.56 | Precision 0.73 | Recall 0.73 | F1-score 0.7 | Precision 0.79 | Recall 0.34 | F1-score 0.88 | Precision 0.79 | Recall 0.24 | F1-score 0.36 | Precision 0.79 |

VI. DISCUSSION AND CONCLUSION

This paper proposes using point pattern features in stark contrast to the commonly used "global feature" for defect detection. A common approach to deal with point pattern features is via transforming these features into global feature using different mapping methods, such as bag of visual words. By contrast, we propose to model point pattern features within...
the random finite set framework. The random finite set framework has been used to estimate the cardinality and density of these features as a sophisticated way to build a statistical model that best fits the normal samples by maximizing the log-likelihood. The main hypothesis here is that when there is a defect, the cardinality and density of point pattern features change, there will be more/fewer edges, and this is captured via random finite set density. Different handcrafted and pre-trained deep point pattern features have been used as feature set measurements to examine which point pattern feature performs better. Experiment on the large-scale defect detection dataset (MVTec-AD) was conducted, and we compared the proposed point features within RFS framework with different deep global-based feature methods and ranked them. Experimental results showed that using feature extraction (especially SIFT) within the RFS framework for defect detection has the best performance (ranked first), due to its ability to capture the strong response along edges. The results indicate that when there is a defect, the number of edges is significantly different. Having said that, the main limitation of using SIFT is the need for manual setting of edge and peak thresholds, which can highly affect the performance.

On the other hand, we observed that the pre-trained deep point feature detection methods do not consistently generate better results. The second-best point pattern feature detection method was r2d2 (average rank is four). This is largely due to the fact that the network generates repeatable and reliable point pattern features descriptors. Furthermore, LF-net did not perform well compared to SIFT, and we think this is due to the network converting the input image into gray-scale and neglecting the contribution of color information. It should also be said that these networks are trained on entirely different domain datasets, which are used for the visual matching task. Although these deep learning methods perform poorly for feature extraction, they can still generate comparative results for some objects when combined with RFS-based defect detection. Finally, the results show that if features are chosen correctly, the RFS-based anomaly detection methods outperform global feature-based models.

ACKNOWLEDGMENT

This work was supported by the Australian Research Council (the ARC) via the Project Linkage grant LP160101081.

REFERENCES

[1] S.-H. Huang and Y.-C. Pan, “Automated visual inspection in the semiconductor industry: A survey,” Computers in Industry, vol. 66, pp. 1–10, 2015.

[2] F.-C. Chen and M. Jahanshahi, “Nb-cnn: Deep learning-based crack detection using convolutional neural network and naive bayes data fusion,” IEEE Transactions on Industrial Electronics, vol. 65, no. 5, pp. 4392–4400, 2017.

[3] R. Ren, T. Huang, and K. C. Tan, “A generic deep-learning-based approach for automated surface inspection,” IEEE Transactions on Cybernetics, vol. 48, no. 3, pp. 929–940, 2017.

[4] J. Wang, Y. Ma, L. Zhang, R. X. Gao, and D. Wu, “Deep learning for smart manufacturing: Methods and applications,” Journal of Manufacturing Systems, vol. 48, pp. 144–156, 2018.

[5] D. Weimer, B. Scholz-Reiter, and M. Shipitani, “Design of deep convolutional neural network architectures for automated feature extraction in industrial inspection,” CIRP Annals, vol. 65, no. 1, pp. 417–420, 2016.

[6] Y.-J. Chu, W. Choi, G. Suh, S. Mahmoudkiani, and O. Buyukkurt, “Autonomous structural visual inspection using region-based deep learning for detecting multiple damage types,” Computer-Aided Civil and Infrastructure Engineering, vol. 33, no. 9, pp. 731–747, 2018.

[7] K. Makantasis, E. Protopapadakis, A. Doula, M. N. Doulamis, and C. Loupos, “Deep convolutional neural networks for efficient vision-based tunnel inspection,” in 2015 IEEE international conference on intelligent computer communication and processing (ICCPP). IEEE, 2015, pp. 335–342.

[8] R. Jenssen, D. Roverso et al., “Automatic autonomous vision-based power line inspection: A review of current status and the potential role of deep learning,” International Journal of Electrical Power & Energy Systems, vol. 99, pp. 107–120, 2018.

[9] L. Zhang, F. Yang, Y. D. Zhang, and Y. J. Zhu, “Road crack detection using deep convolutional neural network,” in 2016 IEEE international conference on image processing (ICIP). IEEE, 2016, pp. 3708–3712.

[10] X. Gubert, V. M. Patel, and R. Chellappa, “Deep multitask learning for railway track inspection,” IEEE transactions on intelligent transportation systems, vol. 18, no. 1, pp. 153–164, 2016.

[11] J. Wang, P. Fu, and R. X. Gao, “Machine vision intelligence for product defect inspection based on deep learning and hough transform,” Journal of Manufacturing Systems, vol. 51, pp. 52–60, 2019.

[12] L. Li, K. Ota, and M. Dong, “Deep learning for smart industry: Efficient manufacture inspection system with fog computing,” IEEE Transactions on Industrial Informatics, vol. 14, no. 10, pp. 4665–4673, 2018.

[13] H. M. Elbehery, A. A. Hefnawy, and M. T. Elewa, “Visual inspection for fired tile’s surface defects using wavelet analysis,” GVIP (G5), vol. 02, pp. 1–8, 2005.

[14] J. Jing, H. Zhang, J. Wang, P. Li, and J. Jia, “Fabric defect detection using gabor filters and defect classification based on lbp and tamura method,” Journal of the Textile Institute, vol. 104, no. 1, pp. 18–27, 2013.

[15] D. G. Lowe, “Distinctive image features from scale-invariant keypoints,” International journal of computer vision, vol. 60, no. 2, pp. 91–110, 2004.

[16] X. Gubert, V. M. Patel, and R. Chellappa, “Robust faster detection for autonomous visual railway track inspection,” in 2015 IEEE Winter Conference on Applications of Computer Vision. IEEE, 2015, pp. 694–701.

[17] D. Shumin, L. Zhoufeng, and L. Chunlei, “Adaboost learning for fabric defect detection based on hog and svm,” in 2011 International conference on multimedia technology. IEEE, 2011, pp. 2903–2906.

[18] K. He, X. Zhang, S. Ren, and J. Sun, “Deep residual learning for image recognition,” in Proceedings of the IEEE conference on computer vision and pattern recognition, 2016, pp. 770–778.

[19] A. Krizhevsky, I. Sutskever, and G. E. Hinton, “Imagenet classification with deep convolutional neural networks,” in Advances in neural information processing systems, 2012, pp. 1097–1105.

[20] W. Dai, A. Mujeed, M. Erdt, and A. Sourin, “Soldering defect detection in automatic optical inspection,” Advanced Engineering Informatics, vol. 43, p. 101004, 2020.

[21] X. Yin, Y. Chen, A. Bouferguene, H. Zaman, M. Al-Hussein, and L. Kurach, “A deep learning-based framework for an automated defect detection system for sewer pipes,” Automation in Construction, vol. 109, p. 102967, 2020.

[22] 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, 2019.

[23] U. Hwang, D. Jung, and S. Yoon, “Hexagan: Generative adversarial nets for real world classification,” arXiv preprint arXiv:1902.09913, 2019.

[24] A. Oliver, A. Odena, C. A. Raffel, E. D. Cubuk, and I. Goodfellow, “Realistic evaluation of deep semi-supervised learning algorithms,” in Advances in Neural Information Processing Systems, 2018, pp. 3235–3246.

[25] J. Yoon, J. Jordon, and M. Van Der Schaar, “Gain: Missing data imputation for autonomous visual railway track inspection,” in 2019 IEEE Conference on Computer Vision and Pattern Recognition.

VOLUME xx, yyyy

This article has been accepted for publication in a future issue of this journal, but has not been fully edited. Content may change prior to final publication. Citation information: DOI 10.1109/ACCESS.2021.3130261, IEEE Access

Author et al.: Point Pattern Feature-Based Anomaly Detection for Manufacturing Defects, in the Random Finite Set Framework
568 B.-N. Vo, N. Dam, D. Phung, Q. N. Tran, and B.-T. Vo, “Model-based learning for point pattern data,” Pattern Recognition, vol. 84, pp. 136–151, 2018.
569 J. Sivic and A. Zisserman, “Video google: A text retrieval approach to object matching in videos,” in multimodal. IEEE, pp. 2003, 1470.
570 D. Carrera, G. Boracchi, A. Foil, and B. Wohlgem, “Detecting anomalous structures by convolutional sparse models,” in 2015 International Joint Conference on Neural Networks (IJCNN). IEEE, 2015, pp. 1–8.
571 C. Huang, J. Cao, F. Ye, M. Li, Y. Zhang, and C. Lu, “Inverse-transform autoencoder for anomaly detection,” 2019.
572 S. Venkataramanan, K.-C. Peng, R. V. Singh, and A. Mahalanobis, “Attention guided anomaly localization in images,” in European Conference on Computer Vision. Springer, 2020, pp. 485–503.
573 S. Akyay, A. Atapour-Abarghouei, and T. P. Brock, “Ganomaly: Semi-supervised anomaly detection via adversarial training,” in Asian conference on computer vision. Springer, 2018, pp. 622–637.
574 D. Li, D. Chen, J. Goh, and S.-K. Ng, “Anomaly detection with generative adversarial networks for multivariate time series,” arXiv preprint arXiv:1809.04758, 2018.
575 C. Yang, T. Tanay, E. J. Morton, and L. D. Griffin, “Transfer representation-learning for anomaly detection.” JMLR, 2016.
576 T. S. Nazare, R. F. De Mello, and M. A. Ponti, “Are pre-trained cnns good feature extractors for anomaly detection in surveillance videos?” arXiv preprint arXiv:1811.08495, 2018.
577 J. Donahue, Y. Jia, O. Vinyals, J. Hoffman, N. Zhang, E. Tzeng, and R. Darrell, “A deep convolutional activation feature for generic visual recognition,” UC Berkeley & ICSI, Berkeley, CA, USA, vol. 1.
578 O. Rippel, P. Mertens, and D. Merhof, “Modeling the distribution of normal data in pre-trained deep features for anomaly detection,” in 2020 IEEE 25th International Conference on Pattern Recognition (ICPR). IEEE, 2021, pp. 6726–6733.
579 M. Tan and Q. Le, “Efficientnet: Rethinking model scaling for convolutional neural networks,” in International Conference on Machine Learning. PMLR, 2019, pp. 6105–6114.
580 I. Golan and R. El-Yaniv, “Deep anomaly detection using geometric transformations,” arXiv preprint arXiv:1805.10917, 2018.
581 L. Bergman and Y. Hoshen, “Classification-based anomaly detection for general data,” in International Conference on Learning Representations, 2019.
582 K. Sohn, C.-L. Li, J. Yoon, M. Jin, and T. Pfister, “Learning and evaluating representations for deep one-class classification,” arXiv preprint arXiv:2011.02578, 2020.
583 J. Tack, S. Mo, J. Jeong, and J. Shin, “Csi: Novelty detection via contrastive learning on distributionally shifted instances,” arXiv preprint arXiv:2007.08176, 2020.
584 J. Deng, W. Dong, R. Socher, L.-J. Li, K. Li, and L. Fei-Fei, “Imagenet: A large-scale hierarchical image database,” in 2009 IEEE conference on computer vision and pattern recognition. Ieee, 2009, pp. 248–254.
585 D. DeTone, T. Malisiewicz, and A. Rabinovich, “Superpoint: Self-supervised interest point detection and description,” in Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition Workshops, 2018, pp. 224–236.
586 R. P. Mahler, Statistical multisource-multitarget information fusion. Artech House Norwood, MA, 2007, vol. 685.
587 A. M. Kamoona, A. K. Gostar, R. Tannkoon, A. Bah-Hadiashar, D. A. Cadiya, J. Thorpe, and R. Hoseinnejad, “Random finite set-based anomaly detection for safety monitoring in construction sites,” IEEE Access, vol. 7, pp. 105 710–105 720, 2019.
588 R. P. S. Mahler, “Multitarget bayes filtering via first-order multitarget moments,” IEEE Transactions on Aerospace and Electronic Systems, vol. 39, no. 4, pp. 1152–1178, 2003.
589 A. M. Kamoona, A. K. Gostar, A. Bah-Hadiashar, and R. Hoseinnejad, “Sparsity-based naive bayes approach for anomaly detection in real surveillance videos,” in 2019 International Conference on Control, Automation and Information Sciences (ICCAIS), 2019, pp. 1–6.
590 B.-T. Vo, B.-N. Vo, and A. Cantoni, “The cardinality balanced multitarget multi-bernoulli filter and its implementations,” IEEE Transactions on Signal Processing, vol. 57, no. 2, pp. 409–423, 2008.
591 R. P. Mahler, Advances in statistical multisource-multitarget information fusion. Artech House, 2014.
592 S. Roenicke, A. No, and K. Dietmayer, “The labeled multi-bernoulli filter,” IEEE Transactions on Signal Processing, vol. 62, no. 12, pp. 3246–3260, 2014.
593 F. Pap, B.-N. Vo, B.-T. Vo, C. Fantacci, and M. Beard, “Generalized labeled multi-bernoulli approximation of multi-object densities,” IEEE Transactions on Signal Processing, vol. 63, no. 20, pp. 5487–5497, 2015.
594 N. Dalal and B. Triggs, “Histograms of oriented gradients for human detection,” in 2005 IEEE computer society conference on computer vision and pattern recognition (CVPR’05), vol. 1. IEEE, 2005, pp. 886–893.
595 K. Mikolajczyk, T. Tuytelaars, C. Schmid, A. Zisserman, J. Matas, F. Schaffalitzky, T. Kadir, and L. Van Gool, “A comparison of affine region detectors,” International Journal of computer vision, vol. 65, no. 1-2, pp. 43–72, 2005.
596 C. Schmid, R. Mohr, and C. Bauckhage, “Evaluation of interest point detectors,” International Journal of computer vision, vol. 37, no. 2, pp. 151–172, 2000.
597 K. Mikolajczyk and C. Schmid, “A performance evaluation of local descriptors,” IEEE transactions on pattern analysis and machine intelligence, vol. 27, no. 10, pp. 1615–1630, 2005.
598 S.-E. Wei, V. Ramakrishna, T. Kanade, and Y. Sheikh, “Convolutional pose machines,” in Proceedings of the IEEE conference on Computer Vision and Pattern Recognition, 2016, pp. 4724–4732.
599 W. Liu, D. Anguelov, D. Erhan, C. Szegedy, S. Reed, C.-Y. Fu, and A. C. Berg, “Ssd: Single shot multibox detector,” in European conference on computer vision. Springer, 2016, pp. 21–37.
600 K. M. Yi, E. Trulls, V. Lepeit, and P. Fua, “Lift: Learned invariant feature transform,” in IEEE Conference on Computer Vision. Springer, 2016, pp. 467–483.
601 Y. Ono, E. Trulls, P. Fua, and K. M. Yi, “Lf-net: learning local features from images,” in Advances in neural information processing systems, 2018, pp. 6234–6244.
602 M. Dusmanu, I. Rocco, T. Pajdla, M. Pollefeys, J. Sivic, A. Torii, and T. Sattler, “D2-net: A trainable cnn for joint detection and detection of local features,” in Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition, 2019, pp. 8092–8101.
603 J. Revaud, P. Weinzaepfel, C. De Souza, N. Pion, G. Szegedy, Y. Cabon, and M. Hurnberger, “R2d2: Repeateable and reliable detector and descriptor,” arXiv preprint arXiv:1906.06195, 2019.
604 C. M. Bishop, Pattern recognition and machine learning. springer, 2006.
605 A. Dai, A. X. Chang, M. Savva, M. Halber, T. Funkhouser, and M. Nießner, “Scannet: Richly-annotated 3d reconstructions of indoor scenes,” in Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition, 2017, pp. 5828–5839.
606 Z. Li and N. Snavely, “Megadepth: Learning single-view depth prediction from internet photos,” in Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition, 2018, pp. 2041–2050.
607 P. Napoletano, F. Piccoli, and R. Schettini, “Anomaly detection in nanofibrous materials by cnn-based self-similarity,” Sensors, vol. 18, no. 1, p. 209, 2018.
608 T. Böttger and M. Ulrich, “Real-time texture error detection on textured surfaces with compressive sensing,” Pattern Recognition and Image Analysis, vol. 26, no. 1, pp. 88–94, 2016.
609 C. Steger, M. Ulrich, and C. Wiedemann, Machine vision algorithms and applications. John Wiley & Sons, 2018.
610 M. W. Milo, M. Roan, and B. Harris, “A new statistical approach to automated quality control in manufacturing processes,” Journal of Manufacturing Systems, vol. 36, pp. 159–167, 2015.
611 Y. Zhao, X. Liu, S. Gan, and W. Zheng, “Predicting disk failures with hmm-and hsmm-based approaches,” in Industrial Conference on Data Mining. Springer, 2010, pp. 390–404.
612
AMMAR KAMOONA was awarded Master degree in electronic and electrical engineering from Swinburne University of Technology, Melbourne, Australia, in 2016. He has been a student member of IEEE since 2013, and the recipient of two certificates of excellence in RF circuit design and Stochastic and Survival analysis from Swinburne University, as well as a golden key certificate for being one of the top achiever students. He worked as Assistant lecturer at Department of Electrical Engineering, University of Kufa, Iraq from 2017 to 2018. Currently, Ammar is a PhD student at RMIT University of technology, Melbourne, Australia. His current research interests include Computer Vision, RFS filters, Robotics and Optimization, and FPGA applications.

AMIRALI KHODADADIAN GOSTAR received his BSc degree in Electrical Engineering and MSc degree in Philosophy of Science, PhD degree in Mechatronics Engineering from RMIT University. He is currently a postdoctoral research fellow in School of Engineering at RMIT University. His research interests include sensor management, data fusion, and multitarget tracking.

ALIREZA BAB-HADIASHAR received his BSc and MEng in Mechanical Engineering, then PhD in Robotics from Monash University. He has held various positions in Monash University, Swinburne University of Technology and RMIT University where he is currently a professor of mechatronics and leads the intelligent automation research group. His main area of research interest is intelligent automation in general, and robust data fitting in machine vision, deep learning for detection and identification, and robust data segmentation, in particular.

REZA HOSEINNEZHAD received his BSc, MSc, and PhD in Electrical Engineering from University of Tehran, Iran, in 1994, 1996 and 2002, respectively. He has held various positions at University of Tehran, Swinburne University of Technology, The University of Melbourne, and RMIT University where he has worked since 2010 and is currently a Professor and Research Development Lead as well as the Discipline Leader (Manufacturing Mechatronics) at School of Engineering. His main areas of research interest are statistical information fusion, random finite sets, multi-object tracking, deep learning, and robust multi-structure data fitting in computer vision.