Optics and Lasers in Engineering 128 (2020) 106000

Contents lists available at ScienceDirect

Optics and Lasers in Engineering

journal homepage: www.elsevier.com/locate/optlaseng

Saliency detection for strip steel surface defects using multiple constraints and improved texture features

Guorong Song [a,b], Kechen Song [a,b,*], Yunhui Yan [a,b,*]

[a] School of Mechanical Engineering and Automation, Northeastern University, Shenyang, Liaoning 110819, China
[b] Key Laboratory of Vibration and Control of Aero-Propulsion Systems Ministry of Education of China, Northeastern University, Shenyang 110819, China

ARTICLE INFO

Keywords:
Saliency detection
Multiple constraints
Texture features
Spectral clustering
Label information propagation

ABSTRACT

Surface defect detection of strip steel is still a challenging task for its complex variances, e.g., intra-class defects exist large differences in appearance while inter-class defects contain similar parts. To address these issues, we regard the defect object as the salient part of the image and propose a novel, effective saliency propagation algorithm based on multiple constraints and improved texture features (MCITF). Firstly, we deliberately design 83-dim texture features that are used to generate label matrix (among which the label information viewed as the important basis of diffusion process) in the framework of multiple-instance learning. Then we resort to Laplacian regularization viewed as smoothness constraint for enlarging the gap between defect objects and background, and high-level prior (background, object, and mid-level feature) constraints for constraining the label information propagation process locally in order to uniformly highlight the complete defect objects while effectively suppress the non-salient background. Finally, we observe that the superpixel segmentation algorithm based on spectral clustering can adequately capture the edge information of defects, thus promoting to yield high-quality pixel-level saliency maps. Experimental results implemented on the real challenging strip steel benchmark database demonstrate that our MCITF model outperforms state-of-the-art methods with large margins and strong robustness in terms of eight evaluation metrics.

1. Introduction

SURFACE defect detection is a crucial step to control the quality of industrial products, especially for strip steel. However, most enterprises mainly utilize manual inspection techniques, which rely heavily on the subjective experience of workers and are easy to yield high missing rate and low efficiency. In recent years, machine vision-based automated defect detection [1–7] methods have received substantial attention due to its high efficiency and accuracy. Among them, the salient object detection as its important branch achieves outstanding performance. The salient model simulates the human visual processing mechanism that captures the subset of vital visual information of image for further processing, and has been widely applied in various computer vision tasks, e.g., defect detection [8,9,45,46] and image segmentation [10].

In the past two decades, enormous representative saliency detection methods have been proposed. Bai et al. [8] utilized the phase-only Fourier transform (POFT) method to obtain the saliency regions, from which the defect objects are extracted by comparing with the defect-free template image. However, this method depends heavily on the quality of template image and POFT can hardly handle images with complex background, which limits its application. Song et al. [9] adopted maximum symmetric surround to output saliency map with well-defined boundary, but it still has low contrast and cannot effectively detect the defects nearby the image border. Achanta et al. [11] introduced a frequency-tuned (FT) algorithm that well preserves the edge information of the salient objects and outputs full resolution saliency map. Later, Cheng et al. [12] designed a histogram-based contrast (HC) method by using color statistical quantization and color space smoothing, which effectively reduce the computational cost. Most recently, low-rank matrix recovery theory based saliency detection models such as structured matrix decomposition (SMD) [13], decomposed the image matrix into two components, i.e., the sparse matrix representing the salient objects and the low-rank matrix representing the background. Besides, Huang et al. [14] automatically generated train samples based on the multiple instance learning (MIL) framework, which avoids the complex, time-consuming manual annotation process and significantly improves the accuracy of the saliency detection.

Although previous saliency detection algorithms (e.g., [11,13,14]) have produced promising results, there still exist several shortcomings.

* Corresponding authors at: School of Mechanical Engineering and Automation, Northeastern University, Shenyang, Liaoning 110819, China.
E-mail addresses: songkc@me.neu.edu.cn (K. Song), yanyh@mail.neu.edu.cn (Y. Yan).

https://doi.org/10.1016/j.optlaseng.2019.106000
Received 4 November 2019; Received in revised form 18 December 2019; Accepted 30 December 2019

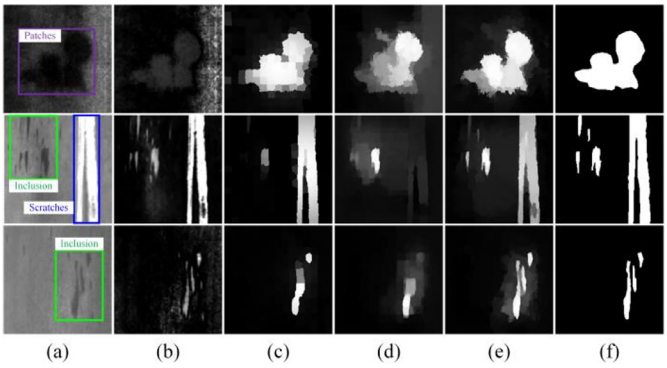

Fig. 1. Typical challenging examples of strip steel surface defects for saliency detection algorithms. From left to right: (a) Source image (b) FT [11] (c) SMD [13] (d) MIL [14] (e) MCITF (ours) (f) Ground truth.

Firstly, the performance of previous method degrades when there are cluttered background or similar appearance between salient object regions and background. As shown in the first row of Fig. 1, FT [11] is difficult to accurately separate the defect objects from the background. Secondly, when dealing with multiple types of defects or scattered objects (the second row of Fig. 1), the previous methods (SMD [13], MIL [14]) cannot highlight the complete defect objects. Thirdly, even dealing with simple image (the third row of Fig. 1), SMD [13] and MIL [14] produce poor saliency maps with blurry boundary. While for FT [11], background noise affects the quality of saliency map.

To address these issues, we propose a saliency propagation algorithm based on multiple constraints and improved texture features (MCITF) model. MCITF can adaptively learn the features of defect objects from complicated background. Meantime, this model fully takes into account the spatial correlation and effectively integrates some prior constraints to promote the generation of satisfying detection results. The general flowchart of our proposed model is shown in Fig. 2.

The main contributions are summarized as follows:

- We formulate the strip steel surface defect detection problem as a saliency value evaluation issue and develop a novel model, i.e., MCITF, for defect object detection. This model effectively combines the advantages of the low-level feature representation and high-level prior knowledge, which are beneficial to enhance the performance of saliency detection.
- We propose an improved local binary pattern (LBP) descriptor, i.e., the median robust adjacent evaluation of LBP (MRAELBP), and build a well-designed 83-dim texture feature bank. Through the framework of multiple-instance learning, we precisely identify the defect features from the complex background for constructing a label matrix that serves as the important basis of our MCITF model.
- From the perspective of label information propagation, we make full use of the generated label matrix to construct an effective saliency propagation algorithm that is seamlessly embedded into the proposed MCITF model. The algorithm fuses structural smoothing, high-level prior (background, object, and mid-level feature) constraints and label matrix into a diffusion function that owns a closed form solution. Compared with other popular propagation models, our method can capture more accurate intrinsic data manifold structure by using interactive regularization, Laplacian regularization and constraining the diffusion process locally. Hence it can precisely retrieval the defect objects from the image, and achieve higher quality saliency map. Detailed model analysis and discussion are presented in Section 3.3.
- We contribute three categories (Inclusion, Patches, and Scratches) consist of total 900 steel surface defect detection images and corresponding pixel-wise binary maps (ground truth), called SD-saliency-900. Furthermore, we find that the spectral clustering based superpixel segmentation algorithm can fully adapt to the rich texture changes of strip steel and capture the precise edge information. Finally, the experimental results generated by our MCITF model are significantly superior to other state-of-the-art methods, which validates the rationality of the proposed model.

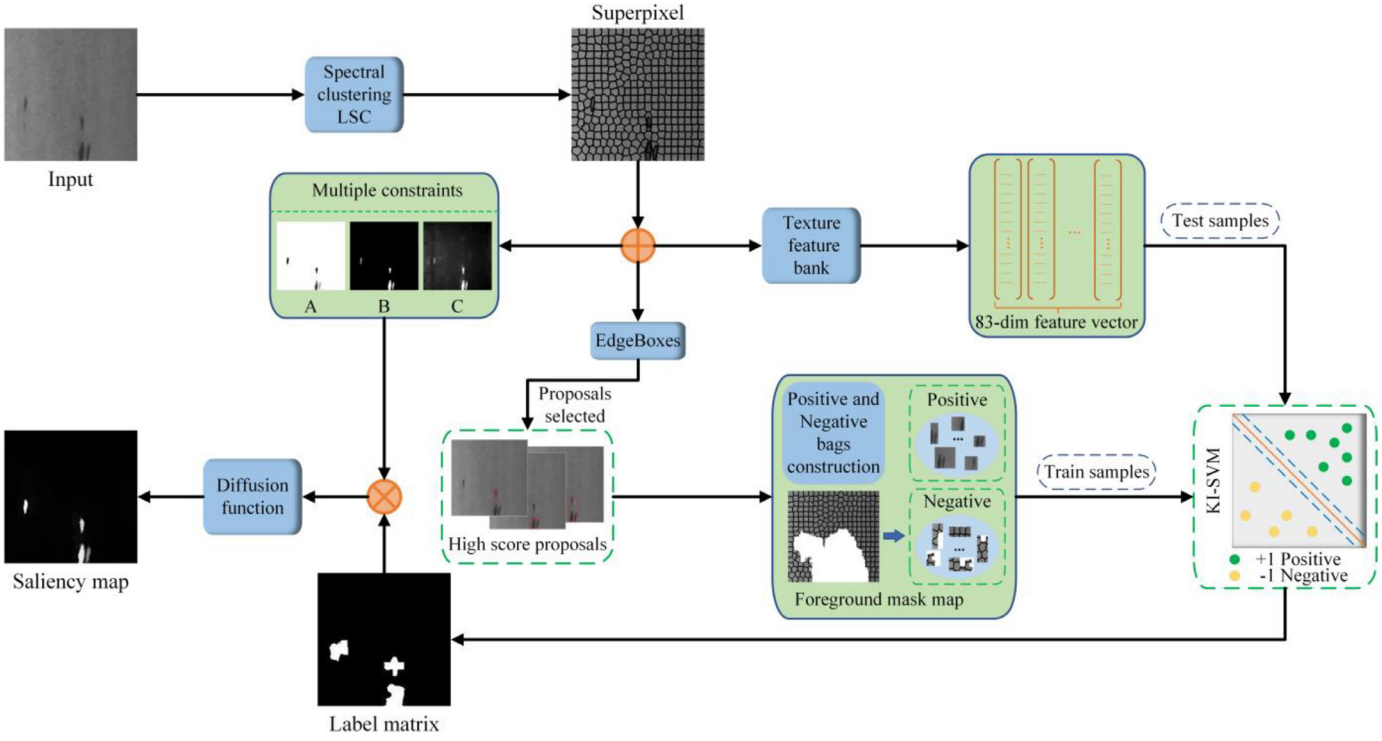

Fig. 2. The flowchart of our proposed MCITF model. Where A means background constraint, B means object constraint, C means mid-level feature constraint.

2. MCITF based saliency propagation algorithm

2.1. Basic model formulation

Given a strip steel surface defect image I, it is first divided into K non-overlapping subregions $\{P_1, P_2, \cdots, P_K\}$ by means of superpixel segmentation. A D-dim feature vector can be extracted from each patch P_i and denoted as $\mathbf{f}_i \in \mathbb{R}^D$. The set of feature vectors constructs a matrix representation of image I, denoted as $\mathbf{F} = [\mathbf{f}_1, \mathbf{f}_2, \cdots, \mathbf{f}_K]^T \in \mathbb{R}^{K\times D}$. Inspired by the model of eye fixation prediction [15], the problem of saliency object detection can be viewed as a task that designs an efficient model. This model can be utilized to obtain a series of potential visual saliency seeds (the region of interest, ROI), of which the labels are taken as the important initial information of the detection model. By making full use of the inherent information of feature matrix $\mathbf{F}$, saliency seeds can be derived from it and then applied to construct a binarized label matrix, denoted as $\{\mathbf{L} = [l_{ij}]_{m\times n}, l_{ij} \in \{1, -1\}\}$. Among the label matrix $\mathbf{L}$, $l_{ij} = 1$ represents the ROI corresponding to defects while $l_{ij} = -1$ represents the redundant information part corresponding to visually consistent background.

However, the saliency detection results acquired only by saliency seeds can hardly achieve the desired experimental performance due to not fully considering the structural information of the image, such as the similarity of adjacent spaces and the consistency of patterns. Thus, some constraints are necessary to limit the diffusion of label information derived from saliency seeds so as to precisely retrieval the defect objects from the image. Besides, constraining the diffusion process locally not only can promote local related superpixels to share similar saliency value, but also can enlarge the gap between salient defect objects and the background. Thus, the saliency propagation model based on constraints can improve the quality of saliency maps and produce competitive results.

Based on the above analysis, we propose a novel MCITF based saliency propagation model as follows:

$$\min_{\mathbf{S}} \Theta(\mathbf{S}, \mathbf{L}) + \mu\Psi(\mathbf{S}) + \mathrm{M}(\mathbf{S}) \tag{1}$$

where $\mathbf{S} = [s_1, s_2, \cdots, s_K]^T \in \mathbb{R}^K$ means the saliency map, among which s_i equals the saliency value of each superpixel. $\mathbf{L}$ is a label matrix. $\Theta(\mathbf{S}, \mathbf{L})$ stands for interactive regularization term to minimize the deviation between $\mathbf{S}$ and $\mathbf{L}$. More specifically, the ROI in $\mathbf{L}$ tends to hold higher saliency values, and vice versa. $\Psi(\cdot)$ is the smoothness constraint term that promotes the generation of continuous saliency values. $\mathrm{M}(\cdot)$ means multiple constraints that fully exploit the structural information of I to uniformly highlight the defect object and effectively suppress the non-salient background. μ is a positive tradeoff parameter.

2.2. Label matrix construction

This stage mainly consists of two parts: the first one focuses on extracting 83-dim deliberately designed feature vector from each superpixel for yielding test samples. While the second one is to build positive and negative bags deemed as the train samples of the key-instance support vector machine (KI-SVM) [16] for getting model parameters according to the current input. Then based on the trained KI-SVM model, binarized label matrix can be obtained by classifying all the test samples.

2.2.1. Texture feature extraction

In this part, we first propose an improved local binary pattern (LBP) descriptor, the median robust adjacent evaluation of LBP (MRAELBP), which owns better discrimination and robustness compared with the vanilla LBP [17]. Then inspired by the work of Jiang *et al.* [18], an 83-dim texture feature bank can be constructed to encode the texture information of image I.

MRAELBP Texture Descriptor: Initially, each image sample I is normalized to be the matrix $\mathbf{X}$ with an average intensity of 128 and a standard deviation of 20, which can weaken the effect of noise to a certain degree [17]. Then we extend the border value of $\mathbf{X}$ using mirror replication to obtain its t-symmetric pad matrix, denoted as $\mathbf{E}_{\mathbf{X}}^t = [e_{ij}]_{(m+2t)\times(n+2t)}$. The central matrix $(\mathbf{E}_{\mathbf{X}}^t)_C$ (same size as $\mathbf{X}$) equals to $\mathbf{E}_{\mathbf{X}}^t$ that excludes all padding values. Later, we introduce the median filter ϕ_w to enhance the robustness of texture descriptor. Let $\mathbf{Z} = \phi_w(\mathbf{E}_{\mathbf{X}}^t)$, $w\times w$ describes the size of the filter window, and similarly, $\mathbf{Z}_C$ represents the central matrix after filter response. Let $t = R + 1$ (R is a sampling radius) and $w = 3$.

As described in [19], the center gray level (C) and the local difference can be used to entirely capture the local structural information of original image. Besides, the local difference can be further decomposed into the sign (S) and the magnitude (M) components. Consequently, three texture operators proposed in MRAELBP are defined as follows:

$$\begin{cases} MRAELBP\_C_{P,R} = s(z_c - \alpha_w), s(x) = \begin{cases} 1, & x \geq 0 \\ 0, & x < 0 \end{cases} \\ MRAELBP\_S_{P,R} = \sum_{p=0}^{P-1} s(a_p - z_c)2^p \\ s_p = s(a_p - z_c), m_p = |a_p - z_c| \\ MRAELBP\_M_{P,R} = \sum_{p=0}^{P-1} t(m_p, \tau)2^p, t(x, c) = \begin{cases} 1, & x \geq c \\ 0, & x < c \end{cases} \end{cases} \tag{2}$$

where $s(x)$ is a sign function. z_c represents the entry of $\mathbf{Z}_C$. The P sampling points are equidistantly distributed around a circle of radius R centered on z_c. If the coordinate of z_c is (0,0), then the coordinate of the pth sampling point is given by $(-R\sin(2\pi p/P), R\cos(2\pi p/P))$. The adjacent estimated gray value of the pth sampling point is set as the average value of its 8-neighborhood (the size of estimation window is 3×3) excluding the value of evaluation center, denoted as a_p. s_p and m_p are two complementary components representing the difference of sign and magnitude, respectively. The threshold α_w is equal to the mean of whole image matrix $\mathbf{Z}_C$, and τ is set as the mean of m_p over the whole image.

In this paper, the encoding scheme of standard (^{riu2}) containing $P + 2$ patterns is applied to validly reduce the impact caused by the uneven illumination and any monotonic gray-scale transformation [17]. In addition, inspired by the combined way of [19], the three operators can be jointly fused to a novel descriptor, which can encode powerful texture feature and own $2(P + 2)(P + 2)$ feature dimension, denoted as MRAELBP_S/M/C. In order to verify the rationality of the proposed MRAELBP descriptor, we carry out a series of classification experiments on Outex_TC_00010 (TC10) [20] database. The classification results presented in Table 1 prove that the performance of MRAELBP is consistently superior or comparable to state-of-the-art texture descriptors, especially for its robustness to salt & pepper (SAP) noise.

Texture Feature Bank: Since superpixels, instead of pixels, used as the basic unit of an image can better preserve the structural information of the image while abstract undesirable details, which can reduce the computational cost. Thus, we first utilize the linear spectral clustering (LSC) superpixel [22] algorithm to partition the image I into K subregions, i.e., $\{P_i | i = 1, 2, \cdots, K\}$, and we discuss the impact of the superpixel segmentation algorithm on the proposed model in Section 3.3. Considering that the SD-saliency-900 database has complex background and rich texture variances, we propose to extract discriminative texture features from superpixel segmentation map motivated on discriminative regional feature integration (DRFI) [18]. These texture features, presented in Table 2 in detail, fully describe the characteristics of image according to three types of regional saliency features, i.e., regional contrast, backgroundness and property descriptor.

In this paper, several dense texture descriptors (encode information on each pixel) including MR8 [23], Schmid [24], Gabor [25] filters plus the proposed MRAELBP descriptor, and superpixels are effectively fused together, which can better grasp the global appearance and local texture changes of the image. The experiment has proved the rationality of the combination strategy. More specifically, MR8 generates 8 (3 scales for 2

Table 1
Comparison of the classification accuracy (%) of the proposed MRAELBP with other state-of-the-art LBP methods on TC10. The reference cited in this table is [21].

(P, R)	(8, 1)					(16, 2)					(24, 3)				
ρ (%)	0	5	10	15	20	0	5	10	15	20	0	5	10	15	20
LBP [17]	84.82	25.86	11.95	9.04	6.56	89.4	21.04	14.43	7.47	4.35	95.08	23.26	7.97	4.17	4.17
CLBP$^{\#}$ [19]	96.56	17.42	17.4	12.19	5.7	98.72	40	7.79	4.82	4.17	98.93	38.28	4.35	4.17	4.17
AECLBP$^{\#}$ [21]	**97.58**	23.02	4.22	4.17	4.17	98.8	22.71	4.24	4.17	4.17	99.19	23.91	7.6	6.9	4.17
MRAELBP$^{\#}$	97.01	**96.41**	**95.78**	**93.57**	**88.2**	**98.93**	**98.7**	**98.23**	**97.24**	**95.13**	**99.22**	**99.19**	**99.11**	**98.91**	**98.1**
SNR (dB)	50	40	30	20		50	40	30	20		50	40	30	20	
LBP [17]	34.66	30.52	13.28	12.47		87.11	84.06	41.28	8.33		93.59	92.27	57.19	8.31	
CLBP$^{\#}$ [19]	92.21	90.52	32.01	8.33		98.33	98.15	65.16	8.15		99.01	98.85	80.96	11.28	
AECLBP$^{\#}$ [21]	97.45	96.93	76.64	15.52		**99.04**	98.78	92.21	25.52		**99.14**	**99.22**	98.2	34.69	
MRAELBP$^{\#}$	**97.55**	**97.4**	**96.15**	**35.6**		98.93	**98.88**	**97.86**	**49.43**		99.09	99.14	**98.91**	**72.06**	

Note that ρ means the level of salt & pepper noise, SNR means the signal-to-noise ratio of Gaussian noise. The superscript '#' means the descriptor with jointly fused '_S/M/C'. The best results are highlighted in blue.

Table 2
Summary of the 83-dim texture feature bank.

Regional contrast & backgroundness descriptor	Computation	Dim	Contrast	Backgroundness
Absolute response of MR8 filters	d	8	$c_1 \sim c_8$	$b_1 \sim b_8$
Absolute response of Schmid filters	d	13	$c_9 \sim c_{21}$	$b_9 \sim b_{21}$
Absolute response of G5 filters	d	5	$c_{22} \sim c_{26}$	$b_{22} \sim b_{26}$
Max response histogram of the G5 & Schmid filters	χ^2	1	c_{27}	b_{27}
Histogram of the MRAELBP feature	χ^2	1	c_{28}	b_{28}
Regional property descriptor	Computation	Dim	Property	
Variances of the response of MR8 filters	*var*	8	$p_1 \sim p_8$	
Variances of the response of Schmid filters	*var*	13	$p_9 \sim p_{21}$	
Variances of the response of G5 filters	*var*	5	$p_{22} \sim p_{26}$	
Variances of the MRAELBP feature	*var*	1	p_{27}	

where d means the L1 norm distance of vectors, denoted as $d(\mathbf{x}_1, \mathbf{x}_2) = \sum_i |x_{1i} - x_{2i}|$; χ^2 means the Chi-square distance of texton response, denoted as $\chi^2(\mathbf{h}_1, \mathbf{h}_2) = \sum_{i=1}^{b}(h_{1i} - h_{2i})^2/(2(h_{1i} + h_{2i}))$ with b being the number of histogram bins; *var* means the variance of all pixels within each superpixel after filter response.

anisotropic filters, plus 2 isotropic) filter bank from the LM [26] set that owns 48 filters; the Schmid set contains 13 rotationally invariant Gabor-like filters; while for Gabor set including 40 filters, we generate 5-dim maximum response across 8 orientations for each scale, denoted as G5, which can not only significantly reduce the dimension of features and the complexity of computation, but also can achieve more discriminative feature descriptions, as in MR8. After getting all the texture feature responses, we can extract 83-dim (2×28+27) feature vector for each sub-region based on the feature calculation strategy of DRFI, viewed as the test samples with initial unknown label, i.e., $\mathbf{F} = [\mathbf{f}_1, \mathbf{f}_2, \cdots, \mathbf{f}_K]^T \in \mathbb{R}^{K\times 83}$.

2.2.2. Bags construction and label generation

Since the construction of label matrix $\mathbf{L}$ plays a fundamental role in the proposed model. To this end, we build reliable positive and negative bags viewed as the train samples of KI-SVM [16] to obtain a powerful discriminative classifier based on the key idea of multiple-instance learning [27]. Then the classifier can be used to attach label to each feature vector (instance) of test samples. It is worth noting that each time the above process is implemented in the single image of the current input. Specifically, we first use Edge Boxes [28] to generate a series of candidate object proposals viewed as the bags and then select high score proposals based on two criteria [14], i.e., discard oversized/undersized boxes and throw out boxes without saliency regions (superpixels with high probability of containing defect objects). Noting that Edge Boxes fails to output object proposals when encountering severe noise disturbance for image I, commonly for SAP noise, so [29] is used to reduce the effect of high-density noise. Later, positive bags can be reasonably constructed according to the high score boxes that are most likely to contain the defect objects, denoted as $Bag^+ = \{\varphi_1, \varphi_2, \cdots, \varphi_{n^+}\}$. φ_i represents the ith bag that owns 83-dim feature vectors, which are extracted from all the superpixels within the corresponding box. While for negative bags, they can be derived from the follows:

$$\begin{cases} Obj(p_i) = \sum_{j=1}^{N_s} Q(\kappa_j)\eta(p_i \in \kappa_j) \\ F_{mask}(p_i) = \begin{cases} 1, & if\ Obj(p_i) \geq T_{obj} \\ Obj(p_i), & otherwise \end{cases} \end{cases} \tag{3}$$

where N_s is the number of selected proposals. $\eta(p_i \in \kappa_j)$ is equal to 1 when the pixel p_i is included in the proposal κ_j and 0 otherwise. $Q(\kappa_j)$ means the objectness score that measures the likelihood of containing defect objects within the bounding box. Thus, we can get the objectness score map Obj and then normalize it to [0, 1]. After that the foreground mask map F_{mask} can be build (see Fig. 2). $T_{obj} = \frac{\beta}{N}\sum_{i=1}^{N} Obj(p_i)$ is a threshold to be the foreground object. β controls the size of the foreground mask. N is the total number of pixels of the image I.

Then we denote $F_{mask}(P_i)$ as the foreground score that is set to the mean of all pixels within the superpixel P_i in the foreground mask map. And a fixed threshold T_{neg} is used to determine if the superpixel is from the background region. Then negative bags can be constructed by grouping all the feature vectors of superpixels whose $F_{mask}(P_i)$ value is smaller than T_{neg}. Finally, we take the positive bags (at least one positive instance in each bag) labeling +1 and the negative bags (all its instances are negative) labeling -1 as the train samples of Bag-KI-SVM [16] to obtain an optimal classifier for the current input. The decision boundary of the classifier can be obtained by correctly dividing the key instances (the most positive instance or ROI obtained from each positive bag) into the positive half space, and all instances in the negative bags otherwise. While the rest instances in the positive bags have little effect on the decision boundary. Besides, the decision boundary of the classifier also needs to maximize the margin between key instances and negative instances (see Fig. 2). Then we use the trained classifier to classify the test samples for getting the estimated label for each superpixel, denoted as $\mathbf{Y} = [y_1, y_2, \ldots, y_K]^T$, $y_i \in \{\pm 1\}$. In essence, $\Theta(\mathbf{S}, \mathbf{L}) = \Theta(\mathbf{S}, \mathbf{Y})$. In other

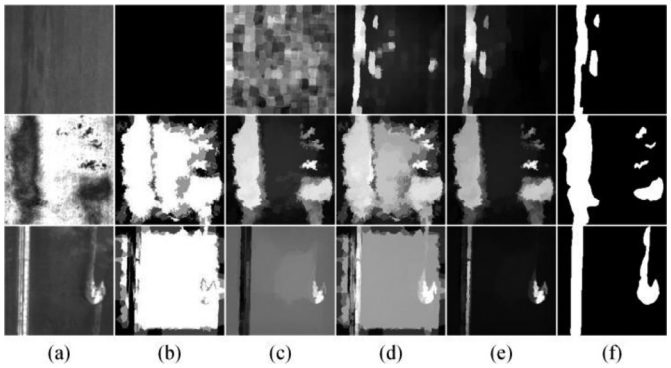

Fig. 3. The performance analysis of saliency detection using different constraints. (a) Source image (b) No constraint (c) Only smoothness constraint (d) Only multiple constraints (e) Full constraints (f) Ground truth.

words, $\mathbf{Y}$ is the vector form of the label matrix $\mathbf{L}$. Then the interactive regularization term can be defined as follows:

$$\Theta(\mathbf{S},\mathbf{Y}) = \sum_{i=1}^{K}(s_i - y_i)^2 = (\mathbf{S}-\mathbf{Y})^T(\mathbf{S}-\mathbf{Y}) \tag{4}$$

2.3. Smoothness constraint

As shown in Fig. 3(b), when the surface defect images of strip steel have similar appearance among background and defect object, confused and scattered defect, jumbled and cluttered background, then label matrix may be difficult to distinguish defect from the background or even fails to obtain correct label information that are taken as the good foundation of subsequent diffusion process. To address this issue, we resort to a Laplacian regularization regarded as a smoothness constraint i.e., if two spatially adjacent image patches have similar intrinsic geometry distribution, e.g., textures and colors, the representations of these two patches should be also close to each other, and vice versa. Simply put, adjacent subregions are more likely to share similar saliency values. Thus, the Laplacian regularization can be defined as follows:

$$\Psi(\mathbf{S}) = \frac{1}{2}\sum_{i,j=1}^{K} w_{i,j}^{(c)}(s_i - s_j)^2 \tag{5}$$

where s_i denotes the saliency value of ith superpixel P_i. $w_{i,j}^{(c)}$ is the (i, j)th element of the affinity matrix $\mathbf{W} \in \mathbb{R}^{K\times K}$ and denotes the weight reflecting the feature similarity between patches P_i and P_j. Besides, an undirected weighted graph model G can be built to describe the surface defect image I. Each superpixel is considered as a node, and the ensemble of edge connecting the pair of nodes constitutes the affinity matrix $\mathbf{W}$, defined as follows:

$$w_{i,j}^{(c)} = \begin{cases} \exp\left(-\frac{||\mathbf{c}_i - \mathbf{c}_j||_2^2}{2\sigma_c^2}\right), & if\ (P_i, P_j) \in \mathbb{N} \\ 0, & otherwise \end{cases} \tag{6}$$

where $\mathbf{c}_i$ denotes the average color value of all pixels within superpixel P_i. σ_c is a parameter to balance the degree of color similarity. $\mathbb{N}$ represents the set of nearby superpixel pairs, i.e., the direct neighbors (first-order reachable) and the neighbors of the direct neighbors (second-order reachable) in the model G.

Since the superpixel segmentation algorithm inevitably splits the perceptually meaningful object region into many small blocks, hence the compactness may be highly distorted. Then the saliency value originally assigned to the defect object can be misallocated to the non-salient background, and vice versa. It may lead to the generation of a poor saliency map that fails to uniformly highlight the complete defect objects and has blurry boundary. Aiming to enhance the performance of final detection, FNCut [30] is applied to cluster over-segment superpixels into larger blocks. It can further reduce intra-class differences and better maintain the local compactness and consistency of spatially adjacent regions due to integrating more geometric structure information. Then the new affinity matrix $\mathbf{V}$ can be defined as follows:

$$v_{i,j} = w_{i,j}^{(c)} + b_{i,j} \tag{7}$$

where $b_{i,j}$ is set to 1 when P_i and P_j are within the same large block after using FNCut, and 0 otherwise.

Based on the above discussion, the smoothness constraint term can be reformulated as follows:

$$\Psi(\mathbf{S}) = \frac{1}{2}\sum_{i,j=1}^{K} v_{i,j}(s_i - s_j)^2 = \mathbf{S}^T\mathbf{L}_\mathbf{M}\mathbf{S} \tag{8}$$

where $\mathbf{L_M}$ is a Laplacian matrix, denoted as $\mathbf{L_M} = \mathbf{D_v} - \mathbf{V}$. $\mathbf{D_v} = diag\{d_{11}, d_{22}, \cdots, d_{KK}\}$ is degree matrix with $d_{ii} = \sum_j v_{i,j}$.

As shown in Fig. 3(c), the saliency maps are significantly improved and possess continuous saliency values by introducing the smoothness constraint. In essence, Laplacian regularization can enlarge the gap between different categories, i.e., foreground object and background [13]. Therefore, patches of the same semantic area are more likely to hold similar or identical saliency value, while patches of heterogeneous areas are own different saliency value.

2.4. High-level prior constraints

Considering that the high-level prior knowledge based on human perception or experience takes into account much more inherent structures and latent properties of images, it can be used to yield superior results. Thus, the superpixel P_i with large foreground probability tends to take a big value s_i (close to 1), while P_i from homogeneous background tends to be suppressed with a small value s_i (close to 0). Then motivated by the work of Shen and Wu [31], we propose to integrate three types of high-level prior constraints (background, object, and mid-level feature constraint) to further boost the performance of the saliency detection.

2.4.1. Background constraint

Initially, the boundary connectivity value of each superpixel P_i in the image I can be computed by [32], then the background constraint term $\mathbf{M_{bg}}$ can be defined as follows:

$$\mathbf{M_{bg}} = \sum_{i=1}^{K} q_i s_i^2 = \mathbf{S}^T\mathbf{D_q}\mathbf{S} \tag{9}$$

where q_i is the background probability mapped from the boundary connectivity value. It describes how likely the superpixel P_i is from the background. $\mathbf{D_q} = diag\{q_1, q_2, \cdots, q_K\}$.

2.4.2. Object constraint

Aiming to make full use of the background probability acquired by the highly reliable boundary connectivity value, an enhanced contrast (called background weight contrast) is further proposed in [32], which reflects the uniqueness and rarity of elements (defect objects). Similarly, the object constraint term $\mathbf{M_{oj}}$ can be defined as follows:

$$\mathbf{M_{oj}} = \sum_{i=1}^{K} u_i(s_i - 1)^2 = (\mathbf{S}-\mathbf{1})^T\mathbf{D_u}(\mathbf{S}-\mathbf{1}) \tag{10}$$

where u_i is the enhanced contrast reflecting the uniqueness of elements. It assigns a larger weight to the salient defect objects compare to the background. $\mathbf{D_u} = diag\{u_1, u_2, \cdots, u_K\}$, and $\mathbf{1} = [1, 1, \cdots, 1]^T \in \mathbb{R}^K$ denotes a one vector.

2.4.3. Mid-level feature constraint

Noting that scattered defect objects or cluttered background may result in inaccurate and incomplete detection, i.e. loss information. To address this problem, we further resort to the mid-level feature cue [33]. Then the mid-level feature constraint term can be defined as follows:

$$\mathbf{M_f} = \sum_{i=1}^{K} h_i(s_i - 1)^2 = (\mathbf{S}-\mathbf{1})^T \mathbf{D_h}(\mathbf{S}-\mathbf{1}) \tag{11}$$

where h_i is the mid-level feature cue and normalized to [0, 1]. It fully considers both color similarity and spatial distribution that can effectively restrain the scattered defect objects, so that they have a high probability to share similar saliency value. $\mathbf{D_h} = diag\{h_1, h_2, \ldots, h_K\}$. $\mathbf{M_f}$ indicates that the superpixel P_i with a larger value h_i is mostly from the foreground region, which assigns larger saliency value that is close to 1.

2.5. Saliency assignment

Based on the above analysis, we integrate the three types of high-level prior constraints by a weighted combination, so the multiple constraints can be defined as follows:

$$\mathrm{M}(\mathbf{S}) = \gamma \mathbf{M_{bg}} + \theta \mathbf{M_{oj}} + \lambda \mathbf{M_f} \tag{12}$$

where γ, θ, λ are positive penalty parameters. Then the model defined in (1) can be regarded as a regression optimization problem, and the diffusion function can be defined as follows:

$$\begin{aligned}\min_{\mathbf{S}} \; & (\mathbf{S}-\mathbf{Y})^T(\mathbf{S}-\mathbf{Y}) + \mu \mathbf{S}^T \mathbf{L_M} \mathbf{S} + \gamma \mathbf{S}^T \mathbf{D_q} \mathbf{S} \\ & + \theta(\mathbf{S}-\mathbf{1})^T \mathbf{D_u}(\mathbf{S}-\mathbf{1}) + \lambda(\mathbf{S}-\mathbf{1})^T \mathbf{D_h}(\mathbf{S}-\mathbf{1})\end{aligned} \tag{13}$$

Taking derivative of the objective function in (13) with respect to $\mathbf{S}$, and let the function to be zero, therefore

$$\mathbf{S} - \mathbf{Y} + \mu \mathbf{L_M} \mathbf{S} + \gamma \mathbf{D_q} \mathbf{S} + \theta \mathbf{D_u}(\mathbf{S}-\mathbf{1}) + \lambda \mathbf{D_h}(\mathbf{S}-\mathbf{1}) = 0 \tag{14}$$

The closed form solution is

$$\mathbf{S}^* = \left(\mathbf{I} + \mu \mathbf{L_M} + \gamma \mathbf{D_q} + \theta \mathbf{D_u} + \lambda \mathbf{D_h}\right)^{-1} (\mathbf{Y} + \theta \mathbf{U} + \lambda \mathbf{H}) \tag{15}$$

where $\mathbf{U} = [u_1, u_2, \cdots, u_K]^T$,$\mathbf{H} = [h_1, h_2, \cdots, h_K]^T$, $\mathbf{I}$ is the identity matrix. Then according to $\mathbf{S}^*$, all pixels within each superpixel will share the same saliency value, thus, saliency values can be assigned to the whole image. Fig. 3(e) intuitively demonstrates the rationality of multiple constraints. Based on the proposed model, we generate a pixel-level saliency map with well-defined object boundary, instead of the vague patch-level counterpart, as shown in Fig. 1(c).

3. Experiment

3.1. Experimental setup

We collected three kinds of typical defects as the benchmark database (i.e., SD-saliency-900) in our experiments, including Inclusion, Patches, and Scratches. In this database, each type of defect contains 300 images (the original resolution is 200×200 pixel). Then we contributed the corresponding pixel-wise binary maps, which are generated by the open annotation tool: LabelMe. Furthermore, the proposed model is qualitatively and quantitatively compared with eleven state-of-the-art saliency detection algorithms, including RCRR [34], DSC [35], WMR [36], 2LSG [37], JUD [38], HC [12], FT [11], SLSM [9], BC [32], MIL [14] and SMD [13].

3.1.1. Evaluation metrics

For a more comprehensive evaluation, eight metrics are used in the experiment, i.e., the precision-recall (PR) curve, the receiver operating characteristic (ROC) curve, the F-measure curve, area under the ROC curve (AUC), mean F-measure (MF), equal error rate (EER), mean absolute error (MAE) and structure-measure (SM). The ROC curve can be created by using false positive rates and true positive rates that are acquired when calculating the PR curve. EER is the value at which false positive rate equals false negative rate. F-measure is a comprehensive evaluation measurement computed by the weight harmonic mean of precision and recall, defined as $F_\eta = \frac{(1+\eta^2)\text{Precision}\times\text{Recall}}{\eta^2\text{Precision}+\text{Recall}}$, where η^2 is set to 0.3 to attach more importance on precision [11]. Besides, MAE [39] measures the dissimilarity between the saliency map S_m and the ground truth G_t. In order to get more reliable evaluation result, we use the recently proposed SM metric [40], which simultaneously considers the region-aware and object-aware structural similarity between S_m and G_t.

3.1.2. Parameter settings and implementation details

In the MRAELBP feature extraction, it is recommended to set the sampling radius R and the sampling points P to 2 and 16, respectively. In the bag construction, the top 70 high score proposals are used as positive bags from the chosen bounding boxes. Then the parameter β and the initial threshold T_{neg} used to construct negative bags are empirically set to 0.8 and 0.4, respectively. In smoothness constraint, we set $\sigma_c^2 = 0.05$ in (6), and the number merged into larger blocks is set to 25 in (7).

In addition, to handle the scale problem, multiscale superpixel segmentation is applied to better fit the appearance changes of defect objects. It also avoids the generation of undesirable block artifacts usually caused by saliency detection at single scale. To this end, coarse saliency maps are generated via three layers of superpixels with different granularities, i.e., $K = 150, 250, 350$, respectively. Then we simply average these saliency maps and normalize to get the final result.

For the final saliency detection model, we set $\mu = 5$ and $\lambda = 1$ to facilitate the generation of continuous and compact saliency maps. Later, we analyzed the impact of the changes of the main parameters θ and γ on the model, and let $\xi = \theta/\gamma$ for easy to observe. The effects of tuning parameters are shown in Fig. 4. We find that the performances of SM and MF increase rapidly before ξ reaches 2, and then flatten as ξ crosses 4. But for MAE, the performance initially decreases, reaches an extremum within the interval of [2, 4] and then increases. Thus, our model is less sensitive to the changes of parameters and works well in the range of [2, 4]. In this paper, we set ξ to 4, i.e., let $\theta = 4$ and $\gamma = 1$. For other

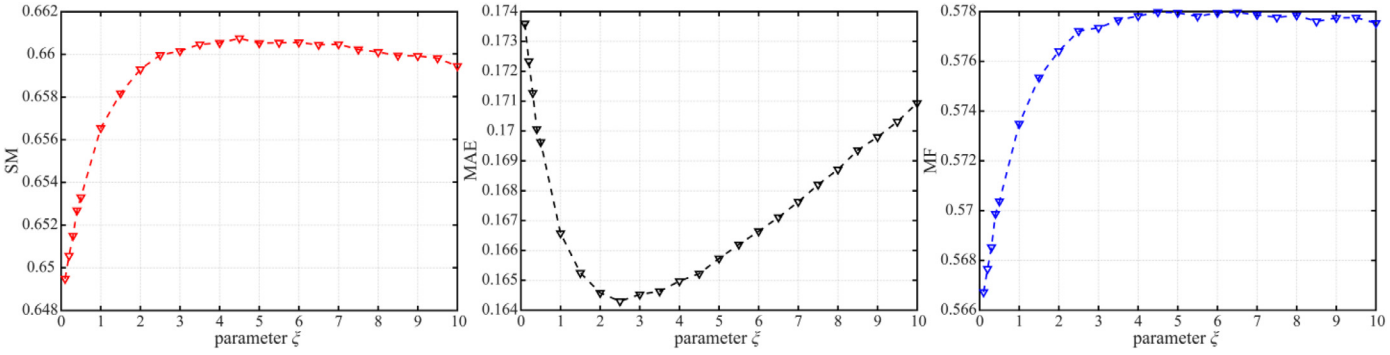

Fig. 4. The impact analysis of parameter ξ.

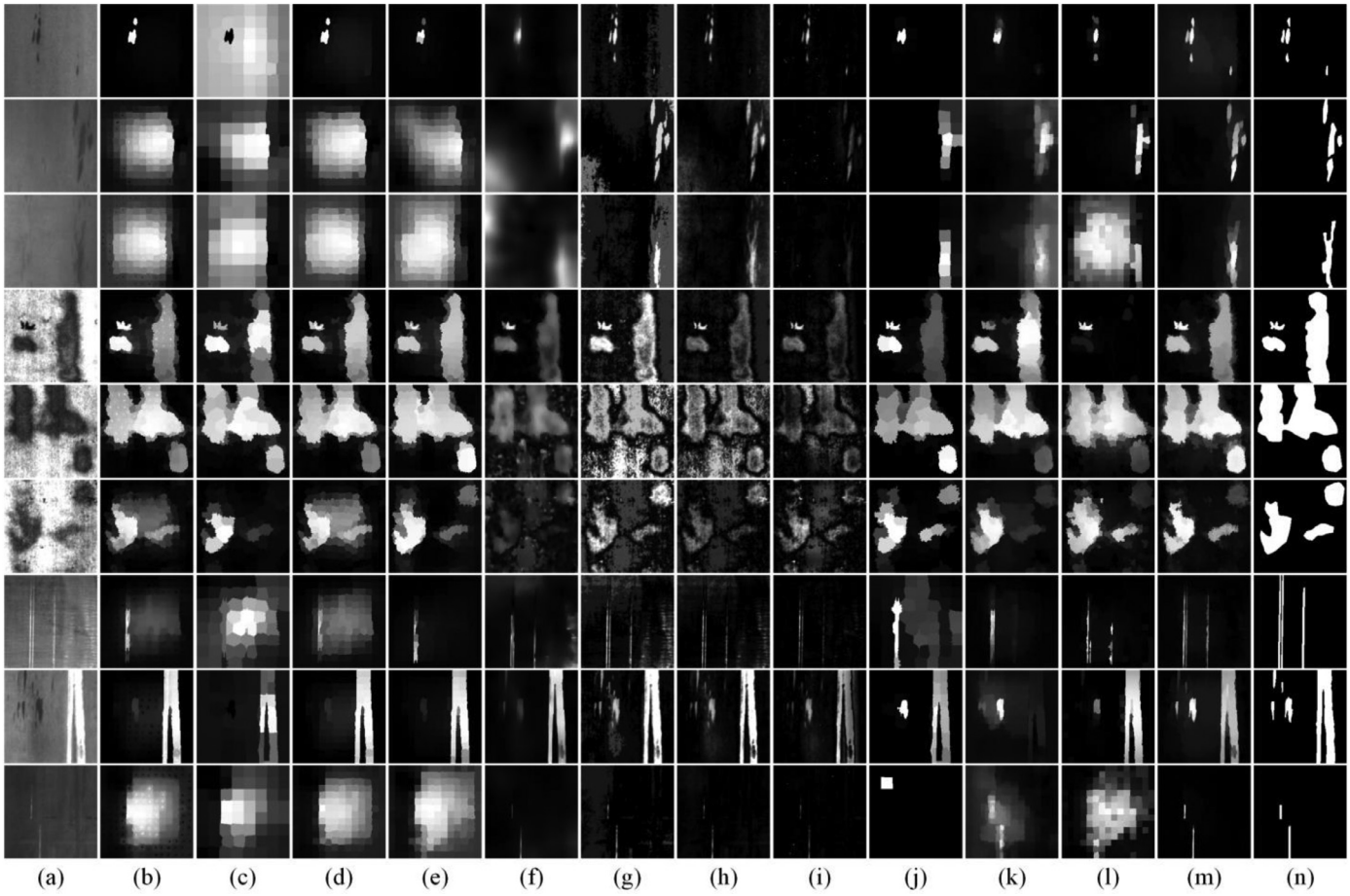

Fig. 5. Visual comparison of saliency maps. (a) Source image (b) RCRR [34] (c) DSC [35] (d) WMR [36] (e) 2LSG [37] (f) JUD [38] (g) HC [12] (h) FT [11] (i) SLSM [9] (j) BC [32] (k) MIL [14] (l) SMD [13] (m) MCITF (ours) (n) Ground truth.

methods in our comparison, we use the source code or executable file provided by the authors with default parameters. In addition, our code and the whole experimental results of all methods are available at *MCITF homepage* (https://sites.google.com/view/mcitf/home).

3.2. Comparison with state-of-the-art methods

3.2.1. Visual comparison

For a visual comparison, Fig. 5 shows some results generated by our MCITF model as well as other best methods in the experiments. We observe that our MCITF model can precisely extract and uniformly highlight the entire defect objects with clear contour for simple images with compact background and distinct defect objects (e.g., row 2, 4 and 5). Some methods (such as FT [11], HC [12] and JUD [38]) also achieve sound saliency maps but contain many noisy results. For images with scattered multiple objects or multiple defects (e.g., row 1, 7 and 8), RCRR [34], WMR [36], 2LSG [37], SMD [13], MIL [14] and BC [32] miss detecting parts of the defect objects, while DSC [35] incorrectly treat part of the background as defects. By contrast, MCITF can locate defect objects with decent accuracy and pop out all of them. For images with low contrast or sharing similar parts between background and defect objects (e.g., row 3 and 9), almost all contrast methods fail or exist fake defects. And the contrast of their saliency maps is low and ambiguous even for JUD [38], FT [11] and SLSM [9]. However, MCITF can effectively identify defect objects from similar background and generate high contrast saliency maps owing to the improved texture features scheme. Finally, for images with cluttered background or heterogeneous defects (e.g., row 6 and 7), RCRR [34], WMR [36] and SLSM [9] assign high saliency on some background region, while BC [32], SMD [13] and MIL [14] have fuzzy contour. In contrast, our model effectively suppresses the background region and achieve high-quality saliency maps by introducing multiple constraints. The above results show that our MCITF model has strong robustness adapted to various complex situations, and verify the rationality of the combination strategy of multiple constraints and improved texture features.

3.2.2. Quantitative comparison

Fig. 6 shows that our MCITF model consistently achieves the best performance among those competitive methods with large margins in terms of PR and ROC curves. While for F-measure curve, the metric score of MCITF is less impressive at low threshold, but it outperforms all the rest methods at high threshold over a wide range. These results objectively prove that our model generates the overall better quality of saliency maps. Then we further consider the performance of the proposed model under severe noise disturbance, presented at *MCITF homepage* due to the page limit. It turns out that, compared with the top three competitive methods (i.e., SMD [13], MIL [14] and BC [32]), MCITF still significantly superior against them, which effectively validates the strong robustness of the proposed model. The quantitative evaluation metrics are presented in Table 3 in detail. We observe that MCITF acquires the best values of SM, AUC and EER in all cases. But for MF, MCITF is slightly lower than the best results achieved by SMD [13] or BC [32] in the case of SAP noise disturbance. Besides, our method also achieves comparable performance to those best results in terms of MAE.

3.2.3. Analysis of time and computation complexity

Table 4 summarizes the average running time of different methods, performed on the SD-saliency-900 dataset using a computer with

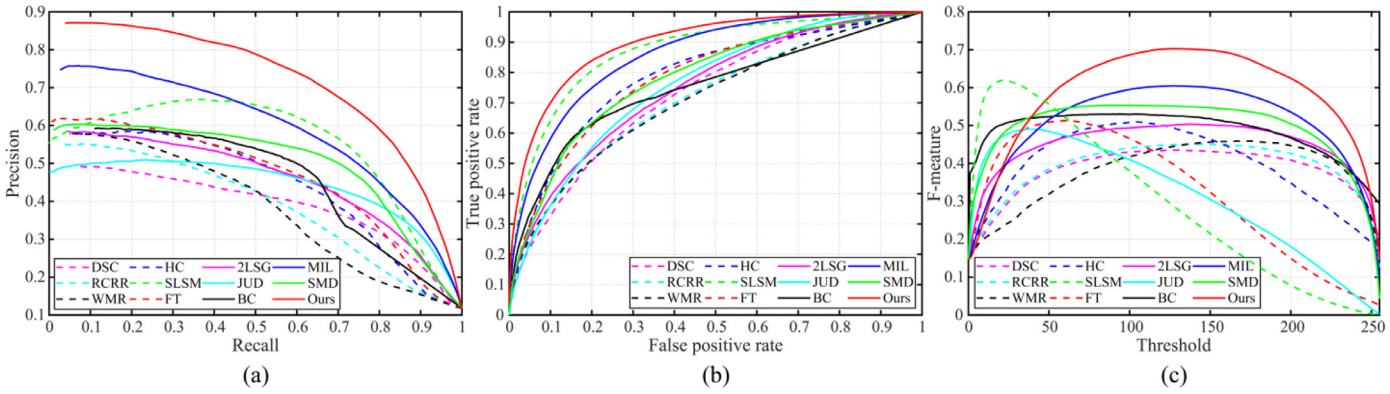

Fig. 6. Quantitative comparison of the proposed model with other competitive algorithms. From left to right: (a) precision-recall (PR) curve (b) receiver operating characteristic (ROC) curve (c) F-measure curve.

Table 3
The results of quantitative evaluation metrics

Noise level	Metric	RCRR [34]	DSC [35]	WMR [36]	2LSG [37]	JUD [38]	HC [12]	FT [11]	SLSM [9]	BC [32]	MIL [14]	SMD [13]	**Ours**
ρ=0	MAE↓	0.2439	0.3045	0.2626	0.2474	0.1737	0.1748	**0.1568**	**0.1127**	**0.1554**	0.1824	0.2045	**0.1650**
	MF↑	0.3899	0.3787	0.3778	0.4463	0.3127	0.3944	0.3204	0.2942	**0.4846**	**0.5035**	**0.4928**	**0.5778**
	SM↑	0.5342	0.5155	0.5204	0.5518	0.5243	0.5751	0.5487	0.5526	**0.5942**	**0.6182**	**0.5838**	**0.6606**
	AUC↑	0.7145	0.7262	0.7112	0.7490	0.7571	**0.7881**	0.7842	**0.8721**	0.7484	**0.8577**	0.7830	**0.8968**
	EER↓	0.3449	0.3334	0.3490	0.3231	0.3108	**0.2657**	0.2806	**0.1972**	0.3021	**0.2248**	0.2828	**0.1799**
ρ=5%	MAE↓	0.2417	0.3205	0.2599	0.2537	**0.1310**	0.2225	**0.1665**	**0.1386**	**0.1494**	0.2390	0.2045	0.2406
	MF↑	0.3810	0.3716	0.3766	**0.4364**	0.1711	0.0936	0.0653	0.1150	**0.4878**	0.4059	**0.4864**	**0.4680**
	SM↑	0.5358	0.5064	0.5213	0.5426	0.4944	0.4145	0.4268	0.4923	**0.5945**	**0.5576**	**0.5837**	**0.5810**
	AUC↑	0.7225	0.7014	0.7095	0.7366	**0.7731**	0.4952	0.4926	0.7375	0.7485	**0.8000**	**0.7830**	**0.8335**
	EER↓	0.3399	0.3546	0.3509	0.3324	0.3090	0.5045	0.5086	0.3306	**0.3016**	**0.2775**	**0.2841**	**0.2461**
ρ=10%	MAE↓	0.2552	0.3536	0.2661	0.2587	**0.1298**	0.2471	**0.1678**	**0.1503**	**0.1519**	0.2456	0.2027	0.2251
	MF↑	0.3799	0.3654	0.3729	**0.4354**	0.1528	0.0889	0.0630	0.0847	**0.4869**	0.3918	**0.4828**	**0.4686**
	SM↑	0.5302	0.4911	0.5164	0.5368	0.4881	0.4136	0.4269	0.4747	**0.5881**	**0.5556**	**0.5785**	**0.5893**
	AUC↑	0.7158	0.6862	0.6989	0.7269	**0.7724**	0.4978	0.4962	0.6936	0.7403	**0.8016**	**0.7740**	**0.8362**
	EER↓	0.3432	0.3699	0.3587	0.3396	**0.3075**	0.5017	0.5049	0.3604	0.3094	**0.2779**	**0.2938**	**0.2467**
ρ=15%	MAE↓	0.2702	0.3687	0.2686	0.2612	**0.1298**	0.2734	**0.1689**	**0.1581**	**0.1678**	0.2479	0.2011	0.2189
	MF↑	0.3735	0.3599	0.3674	**0.4289**	0.1422	0.0913	0.0614	0.0708	**0.4541**	0.3680	**0.4809**	**0.4703**
	SM↑	0.5228	0.4823	0.5134	0.5326	0.4837	0.4108	0.4272	0.4639	**0.5727**	**0.5498**	**0.5753**	**0.5904**
	AUC↑	0.7081	0.6809	0.6947	0.7235	**0.7620**	0.4998	0.4977	0.6609	0.7318	**0.7950**	**0.7653**	**0.8366**
	EER↓	0.3481	0.3725	0.3594	0.3462	0.3159	0.4997	0.5031	0.3821	**0.3145**	**0.2819**	**0.3034**	**0.2442**
ρ=20%	MAE↓	0.2842	0.3892	0.2678	0.2619	**0.1308**	0.2987	**0.1703**	**0.1646**	**0.1753**	0.2494	0.2006	0.2205
	MF↑	0.3595	0.3488	0.3633	**0.4258**	0.1309	0.0946	0.0599	0.0653	**0.4398**	0.3285	**0.4678**	**0.4640**
	SM↑	0.5147	0.4678	0.5135	0.5341	0.4794	0.4077	0.4270	0.4567	**0.5623**	**0.5353**	**0.5701**	**0.5890**
	AUC↑	0.6975	0.6709	0.6917	0.7217	**0.7509**	0.5008	0.4987	0.6352	0.7277	**0.7753**	**0.7577**	**0.8393**
	EER↓	0.3576	0.3797	0.3639	0.3425	0.3215	0.4995	0.5015	0.3996	**0.3163**	**0.2986**	**0.3093**	**0.2410**

[1]The down-arrow ↓ means the smaller value obtained, the better performance, while the up-arrow ↑ means
[2]The best four results are highlighted in blue, green, red and purple, respectively.

Table 4
comparison of running time of different methods

Method	RCRR	DSC	WMR	2LSG	JUD	HC	FT	SLSM	BC	MIL	SMD	Ours	**Ours***
Time(s)	1.095	2.537	1.748	0.639	0.09	1.144	0.126	0.055	0.054	24.134	0.319	30.366	4.927
Code	M	M+C	M+C	M	M+C	C	C	M	M+C	M+C	M+C	M+C	M+C

Note that **Ours*** means MCITF without FNCut, K=[150,250,350] and using 4-core parallel computation for acceleration. M means the code is written in Matlab. C means the code is written in C++.

Intel Core i7-7700 3.6GHz CPU and 16GB RAM. It can be seen that our method achieves the best saliency detection results at the cost of execution time. In fact, most time consumed by our method is at the pre-processing step, mainly including texture feature extraction (about 28%), getting KI-SVM model parameters (about 38%), spectral clustering FNCut (about 23%) and proposals selection (about 3.4%). But for the final saliency assignment (formula (15)), it only takes less than 0.01s that has small computation complexity. In addition, we further analyze the impact of FNCut and multiscale superpixel segmentation on the detection results, as shown in Fig. 7(a). It indicates that the scale of superpixel segmentation does affect the final performance. Considering to adaptively choose the scale size can further reduce the computational cost, which is our future research work. Besides, the average precision of our method without FNCut (denoted as Ours*) is only 2.33% lower

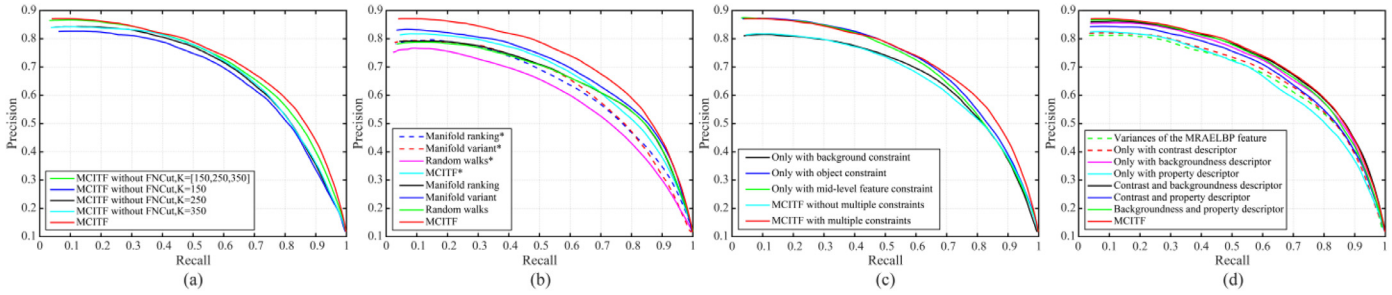

Fig. 7. Comprehensive analysis of saliency detection algorithm based on MCITF model. From left to right: (a) the impact analysis of FNCut and multiscale superpixel segmentation in the proposed MCITF model (b) performance comparison of saliency propagation algorithms based on different models (c) the evaluation of performance contribution for each prior constraint in the proposed MCITF model with respect to PR curve. (d) the performance analysis of different combination descriptors and individual feature. Note that the symbol '*' means models without using multiple constraints.

than that with FNCut, which only slightly weaken the final performance. By utilizing multi-core parallel acceleration calculation, Ours* just takes 4.927s for a typical image, which is acceptable in the real detection process and does not harm the user experiences. In future research, we will further optimize the code to accelerate the proposed saliency propagation algorithm to meet the needs of real-time and high accuracy.

3.3. Model analysis and discussion

3.3.1. Saliency propagation models

To prove the validity of our proposed MCITF based saliency propagation algorithm, we test eight propagation models related to MCITF on the SD-saliency-900 dataset, mainly including popular manifold ranking and its variant [41], and random walks [42]. The optimal experimental performance for each model after separately tuning is shown in Fig. 7(b), we can intuitively draw the following conclusions. (1) The propagation model with multiple constraints is consistently and significantly superior to the one without multiple constraints. (2) Compared with the rest models, the standard random walks* owns the worst performance, but by integrating multiple constraints, it gets a sharp jump in the average precision of 13.92%. (3) The proposed MCITF model achieves the best performance with large margins in terms of PR curve compared with the remaining models, even for MCITF*, which has great advantages and is only slightly lower than the second manifold variant. To objectively explain the above results, we further analyze the underlying reasons. Firstly, all the models listed in Table 5 can be described as such a generic diffusion process based on the core idea of image retrieval [43]: the label information of saliency seeds (query points) assigned a positive ranking score in the label matrix will be propagated to its nearby nodes according to their relevancies to the query. The relevance is defined as the transition probability of a walk from one node to its adjacent node, which is proportional to the affinity value obtained from the weighted graph network. By repeating random walk on the graph until a global stable state is achieved, all the ranking scores (saliency values) of nodes will be obtained, then the detection object (defect) will be retrieved from the image by the diffusion process. Since multiple constraints effectively utilize the prior knowledge, which is helpful to improve the network of the weighted graph model, i.e., assigning larger weights to the pairwise affinity values with great relevance. It also strongly proves that constraining the diffusion process locally is very useful to improve the retrieval scores (the accuracy of defect detection) [43]. Secondly, since the transition matrix P of random walks* reduces the impact of the relevant query point on the current node to a large extent, even if it can highlight the defect object, the quality of saliency map is still severely degraded due to the interference caused by messy background noise. Last but not least, as outlined in the previous section, our initial defined saliency propagation algorithm (MCITF*) can more accurately capture the underlying manifold structure of data. In addition, the detection accuracy is further improved with the effective utilization of multiple constraints. Thus, our proposed MCITF based saliency propagation algorithm yields the best performance of retrieving defect object from the image, and generates more accurate saliency maps with well-defined boundary and full resolution.

3.3.2. Multiple constraints

As shown in Fig. 7(c), we analyze the performance contribution for each prior constraint in the proposed MCITF model in terms of PR curve. It intuitively reflects the fact that any single prior constraint does not achieve the best detection results. Hence the fact indicates that all of these constraints contribute to the final detection accuracy and they complement each other in helping to spread the label information precisely. Even for the background constraint with relatively poor performance, since it constructs a reliable background probability map that is beneficial to improve the adjacent relationship in the weighted graph

Table 5
Different saliency propagation models related to MCITF.

Model	Diffusion function	Closed form solution $\mathbf{S}^*$
Manifold ranking*	$\min_{\mathbf{S}} \sum_{i,j} \frac{1}{2} v_{i,j} \left(\frac{s_i}{\sqrt{d_{ii}}} - \frac{s_j}{\sqrt{d_{jj}}} \right)^2 + \eta \sum_i (s_i - y_i)^2$	$(\mathbf{I} - \alpha\mathbf{N})^{-1}\mathbf{Y}$ where $\alpha = \frac{1}{1+\eta}, \mathbf{N} = \mathbf{D}_{\mathbf{V}}^{-1/2}\mathbf{V}\mathbf{D}_{\mathbf{V}}^{-1/2}$
Manifold variant*	$\mathbf{I} - \alpha\mathbf{N} \approx \mathbf{D}_{\mathbf{V}} - \alpha\mathbf{V}$ where $\approx$ means 'similar to'	$(\mathbf{D}_{\mathbf{V}} - \alpha\mathbf{V})^{-1}\mathbf{Y}$
Random walks*	$\mathbf{S}^{t+1} = \alpha\mathbf{P}^T\mathbf{S}^t + (1-\alpha)\mathbf{Y}$	$(\mathbf{I} - \alpha\mathbf{P}^T)^{-1}\mathbf{Y}$ where $\mathbf{P} = \mathbf{D}_{\mathbf{V}}^{-1}\mathbf{V}$
MCITF*	$\min_{\mathbf{S}} \alpha \sum_{i,j} \frac{1}{2} v_{i,j}(s_i - s_j)^2 + \sum_i (s_i - y_i)^2$	$(\mathbf{I} + \alpha\mathbf{L}_{\mathbf{M}})^{-1}\mathbf{Y}$ where $\mathbf{L}_{\mathbf{M}} = \mathbf{D}_{\mathbf{V}} - \mathbf{V}$
Manifold ranking	$\min_{\mathbf{S}} \sum_{i,j} \frac{1}{2} v_{i,j} \left(\frac{s_i}{\sqrt{d_{ii}}} - \frac{s_j}{\sqrt{d_{jj}}} \right)^2 + \eta \sum_i (s_i - y_i)^2 + \gamma \sum_i q_i s_i^2 + \theta \sum_i u_i (s_i - 1)^2 + \lambda \sum_i h_i (s_i - 1)^2$	$(\mathbf{I} - \alpha\mathbf{N} + \gamma'\mathbf{D}_{\mathbf{q}} + \theta'\mathbf{D}_{\mathbf{u}} + \lambda'\mathbf{D}_{\mathbf{h}})^{-1}[(1-\alpha)\mathbf{Y} + \theta'\mathbf{U} + \lambda'\mathbf{H}]$ where $\gamma' = \gamma\alpha, \theta' = \theta\alpha, \lambda' = \lambda\alpha$
Manifold variant	$\mathbf{I} - \alpha\mathbf{N} \approx \mathbf{D}_{\mathbf{V}} - \alpha\mathbf{V}$	$(\mathbf{D}_{\mathbf{V}} - \alpha\mathbf{V} + \gamma'\mathbf{D}_{\mathbf{q}} + \theta'\mathbf{D}_{\mathbf{u}} + \lambda'\mathbf{D}_{\mathbf{h}})^{-1}[(1-\alpha)\mathbf{Y} + \theta'\mathbf{U} + \lambda'\mathbf{H}]$
Random walks	$\mathbf{I} - \alpha\mathbf{N} \approx \mathbf{I} - \alpha\mathbf{P}^T$	$(\mathbf{I} - \alpha\mathbf{P}^T + \gamma'\mathbf{D}_{\mathbf{q}} + \theta'\mathbf{D}_{\mathbf{u}} + \lambda'\mathbf{D}_{\mathbf{h}})^{-1}[(1-\alpha)\mathbf{Y} + \theta'\mathbf{U} + \lambda'\mathbf{H}]$
MCITF	$\min_{\mathbf{S}} (\mathbf{S}-\mathbf{Y})^T(\mathbf{S}-\mathbf{Y}) + \mu\mathbf{S}^T\mathbf{L}_{\mathbf{M}}\mathbf{S} + \gamma\mathbf{S}^T\mathbf{D}_{\mathbf{q}}\mathbf{S} + \theta(\mathbf{S}-\mathbf{l})^T\mathbf{D}_{\mathbf{u}}(\mathbf{S}-\mathbf{l}) + \lambda(\mathbf{S}-\mathbf{l})^T\mathbf{D}_{\mathbf{h}}(\mathbf{S}-\mathbf{l})$	$(\mathbf{I} + \mu\mathbf{L}_{\mathbf{M}} + \gamma\mathbf{D}_{\mathbf{q}} + \theta\mathbf{D}_{\mathbf{u}} + \lambda\mathbf{D}_{\mathbf{h}})^{-1}(\mathbf{Y} + \theta\mathbf{U} + \lambda\mathbf{H})$

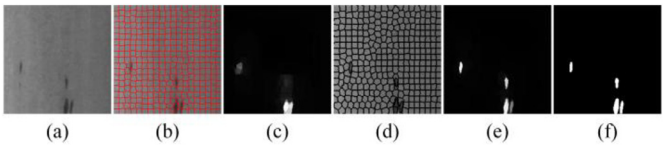

Fig. 8. The effect of different superpixel segmentation algorithms on the proposed saliency detection model. (a) Source image (b) SLIC [44] (c) SLIC based saliency map (d) LSC [22] (e) LSC based saliency map (f) Ground truth.

model, it still obviously superior to the rest competitive saliency detection methods shown in Fig. 6(a). In addition, we further observe that the object constraint performs best, followed by the mid-level feature constraint. Both of them play a major role in the final detection results, and contribute average improvement of 3.73% and 3.13% in precision, respectively.

3.3.3. Superpixel segmentation

We replace the LSC [22] with widely used SLIC [44], and perform MCITF in the same setting to analyze the impact of superpixel segmentation on the proposed model, as shown in Fig. 8. Since the superpixels generated by SLIC (local feature based algorithm) have a great possibility to straddle the background and defect region (see Fig. 8(b)). Thus, the SLIC model generates fuzzy saliency maps with block artifacts. However, by bridging local features and global image structures, LSC is able to generate more reasonable superpixels with high shape compactness and boundary adherence, which adapts well to the changes of image structure and texture. Briefly, LSC tends to generate higher shape irregularity for defect region with abundant texture changes. But for the background with typically smooth change, LSC inclines to produce compact superpixels [22]. Such a trend can be intuitively viewed from Fig. 8(d). The final saliency maps achieved by LSC model not only accurately detect the defect location, but also preserve the defect edge information well. These results indicate that the quality of saliency maps can be further improved by using a reasonable superpixel segmentation algorithm on the proposed model.

3.3.4. Feature extraction

To evaluate the impact of the feature extraction strategy on the proposed model, we replace the improved 83-dim texture features with the 93-dim discriminative regional features proposed in DRFI [18], and perform MCITF in the same setting. The experimental evaluation metrics of MAE, MF, SM and AUC are 0.1796, 0.5605, 0.6414 and 0.8764, respectively. We find that the improved texture feature bank performs consistently and slightly superior to DRFI. It indicates that the improved texture feature bank better adapts to the changes of strip steel database and captures more discriminative features. Besides, we evaluate the performance of different combination descriptors and individual feature to further prove the effectiveness of the well-designed 83-dim fusion features. As shown in Fig. 7(d), it intuitively demonstrates that any individual feature (taking the variance response of the MRAELBP feature as an example) can't effectively distinguish the defect objects from the complex background, thus achieving the overall worst performance. We also observe that our MCITF model is significantly and consistently better than the remaining counterparts using any single feature descriptor. It indicates that all the three types of regional feature descriptors are complementary to each other, and the optimal experimental results can be obtained by integrating their advantages. In addition, the backgroundness and contrast descriptors play a major role in the final detection results, contributing average gains of 3.9% and 2.3% in precision, respectively.

3.3.5. Failure cases

The experimental results demonstrate that the saliency propagation algorithm based on the proposed MCITF model significantly outperforms state-of-the-art saliency detection methods on both subjective and objective evaluation. However, some difficult images still pose challenges to our method as well as those comparative methods, as shown in Fig. 9. We can intuitively see that the source image in the first row has a very low contrast, it is hard to say that there are salient defect objects in the image. That violates the basic assumption of saliency detection, i.e., there are some salient objects that can capture people's attention in the image scene. Thus, our method inevitably generates poor saliency map. In addition, if the salient defect objects are similar with the background regions in terms of texture and color features (see the second row of Fig. 9), our method does not work well, i.e., it cannot completely suppress the undesirable background regions. The underlying reason is that in our proposed model, the construction of label matrix is based on the discriminative power of texture feature bank, while the prior constraints (only show the object constraint for convenience) are mainly considering the color contrast and spatial proximity. In such case, it is difficult to construct a reliable label matrix, and produces poor prior map that gives high weight to the background regions nearby the defect objects. For the third row of Fig. 9, the background regions contained within the defect objects are mistaken as foreground in the object constraint map. The underlying reason is that these background regions have very low boundary connectivity values (i.e., have low probability as background) due to the obstruction of objects on both sides. Therefore, these background regions are also assigned a higher saliency value, which somewhat reduces the quality of the final saliency map. We will work on these problems in the future.

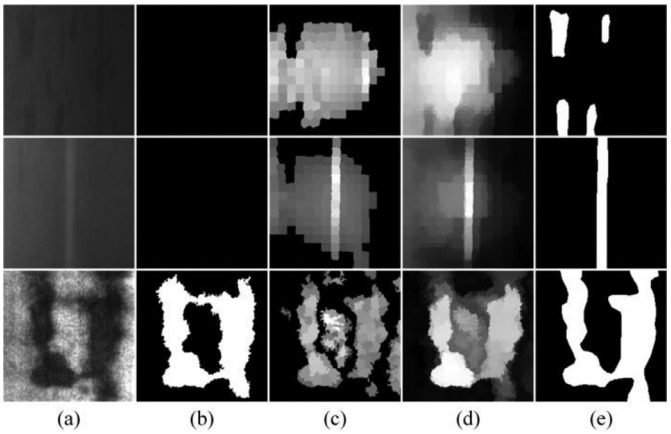

Fig. 9. Some failure examples of our method. (a) Source image (b) Label matrix (c) Object constraint (d) Saliency map (e) Ground truth.

4. Conclusion

In this paper, we formulate the surface defect detection of strip steel as a saliency evaluation problem, and propose a multiple constraints and improved texture features (MCITF) based saliency propagation algorithm in order to achieve high-quality saliency maps, i.e., uniformly highlight the defect objects with well-defined boundary while effectively suppress the saliency value of background. Extensive experiments have proved that our MCITF model performs consistently superior or comparable to other eleven state-of-the-art methods on the SD-saliency-900 database, and has strong robustness. In addition, by considering other useful features (e.g., color, edge, and shape), improving the multi-constraint strategy and constructing effective label information propagation method, our model can be further applied to the surface defect detection of other industrial products, e.g., optical glass and high-speed heavy rail, not limited to strip steel. In the future, we will consider designing an adaptive segmentation algorithm to separate defects from the background based on saliency detection results, and optimize the proposed model to further enhance detection efficiency and accuracy.

Declaration of Competing Interest

The authors declare no conflict of interest.

Acknowledgments

This work is supported by the National Natural Science Foundation of China (51805078, 51374063), the National Key Research and Development Program of China (2017YFB0304200), the Fundamental Research Funds for the Central Universities (N170304014).

References

[1] H. Dong, K. Song, Y. He, J. Xu, et al., " PGA-Net: pyramid feature fusion and global context attention network for automated surface defect detection," IEEE Trans Ind Inform, doi:10.1109/TII.2019.2958826.
[2] Yu H, Li Q, Tan Y, et al. A coarse-to-fine model for rail surface defect detection. IEEE Trans Instrum Meas 2019;68(Mar (3)):656–66.
[3] Song K, Yan Y. A noise robust method based on completed local binary patterns for hot-rolled steel strip surface defects. Appl Surf Sci 2013;285(Nov):858–64.
[4] Fu G, Sun P, Zhu W, et al. A deep-learning-based approach for fast and robust steel surface defects classification. Opt Lasers Eng 2019;121(Oct):397–405.
[5] He D, Xu K, Zhou P, Zhou D. Surface defect classification of steels with a new semi–supervised learning method. Opt Lasers Eng 2019;117(June):40–8.
[6] He Y, Song K, Dong H, Yan Y. Semi-supervised defect classification of steel surface based on multi-training and generative adversarial network. Opt Lasers Eng 2019;122(Nov):294–302.
[7] Luo Q, Sun Y, Li P, et al. Generalized completed local binary patterns for time–efficient steel surface defect classification. IEEE Trans Instrum Meas 2019;68(Mar (3)):667–79.
[8] Bai X, Fang Y, Lin W, Wang L, Ju B. Saliency-based defect detection in industrial images by using phase spectrum. IEEE Trans Ind Inform 2014;10(Nov (4)):2135–45.
[9] Song K, Hu S, Yan Y, Li J. Surface defect detection method using saliency linear scanning morphology for silicon steel strip under oil pollution interference. ISIJ Int 2014;54(11):2598–607.
[10] Zhou S, Wu S, Liu H, Lu Y, Hu N. Double low-rank and sparse decomposition for surface defect segmentation of steel sheet. Appl Sci 2018;8(9):1628.
[11] Achanta R, Hemami S, Estrada F, Susstrunk S. Frequency-tuned salient region detection. In: CVPR; 2009. p. 1597–604.
[12] Cheng M, Mitra NJ, Huang X, Torr PHS, Hu S. Global contrast based salient region detection. IEEE Trans Pattern Anal Mach Intell 2015;37(Mar (3)):569–82.
[13] Peng H, Li B, Ling H, et al. Salient object detection via structured matrix decomposition. IEEE Trans Pattern Anal Mach Intell 2017;39(Apr (4)):818–32.
[14] Huang F, Qi J, Lu H, Zhang L, Ruan X. Salient object detection via multiple instance learning. IEEE Trans Image Process 2017;26(Apr (4)):1911–22.
[15] Wang J, Borji A, Kuo C-CJ, Itti L. Learning a combined model of visual saliency for fixation prediction. IEEE Trans Image Process 2016;25(Apr (4)):1566–79.
[16] Li Y, Kwok JT, Tsang IW, Zhou Z. A convex method for locating regions of interest with multi-instance learning. In: ECML PKDD; 2009. p. 15–30.
[17] Ojala T, Pietikainen M, Maenpaa T. Multiresolution gray-scale and rotation invariant texture classification with local binary patterns. IEEE Trans Pattern Anal Mach Intell 2002;24(July (7)):971–87.
[18] Jiang H, Wang J, Yuan Z, et al. Salient object detection: a discriminative regional feature integration approach. In: CVPR; 2013. p. 2083–90.
[19] Guo Z, Zhang L, Zhang D. A completed modeling of local binary pattern operator for texture classification. IEEE Trans Image Process 2010;19(June (6)):1657–63.
[20] Ojala T, Maenpaa T, Pietikainen M, et al. Outex - new framework for empirical evaluation of texture analysis algorithms. In: ICPR; 2002. p. 701–6.
[21] Song K, Yan Y, Zhao Y, Liu C. Adjacent evaluation of local binary pattern for texture classification. J Visual Commun Image Represent 2015;33(Nov):323–39.
[22] Chen J, Li Z, Huang B. Linear spectral clustering superpixel. IEEE Trans Image Process 2017;26(July (7)):3317–30.
[23] Varma M, Zisserman A. A statistical approach to texture classification from single images. Int J Comput Vis 2005;62(Apr (1)):61–81.
[24] Schmid C. Constructing models for content-based image retrieval. CVPR Kauai, HI; 2001. II-II.
[25] Haghighat M, Zonouz S, Abdel-Mottaleb M. CloudID: trustworthy cloud-based and cross-enterprise biometric identification. Expert Syst Appl 2015;42(Nov (21)):7905–16.
[26] Leung T, Malik J. Representing and recognizing the visual appearance of materials using three-dimensional textons. Int J Comput Vis 2001;43(June (1)):29–44.
[27] Carbonneau M-A, Cheplygina V, Granger E, Gagnon G. Multiple instance learning: a survey of problem characteristics and applications. Pattern Recognit 2018;77(May):329–53.
[28] Zitnick CL, Dollár P. Edge boxes: locating object proposals from edges. In: ECCV; 2014. p. 391–405.
[29] Erkan U, Gökrem L, Enginoğlu S. Different applied median filter in salt and pepper noise. Comput Electr Eng 2018;70(Aug):789–98.
[30] Kim TH, Lee KM, Lee SU. Learning full pairwise affinities for spectral segmentation. IEEE Trans Pattern Anal Mach Intell 2013;35(July (7)):1690–703.
[31] Shen X, Wu Y. A unified approach to salient object detection via low rank matrix recovery. In: CVPR; 2012. p. 853–60.
[32] Zhu W, Liang S, Wei Y, Sun J. Saliency optimization from robust background detection. In: CVPR; 2014. p. 2814–21.
[33] Lu S, Mahadevan V, Vasconcelos N. Learning optimal seeds for diffusion-based salient object detection. In: CVPR; 2014. p. 2790–7.
[34] Yuan Y, Li C, Kim J, Cai W, Feng DD. Reversion correction and regularized random walk ranking for saliency detection. IEEE Trans Image Process 2018;27(Mar (3)):1311–22.
[35] Wu X, Ma X, Zhang J, Wang A, Jin Z. Salient object detection via deformed smoothness constraint. In: ICIP; 2018. p. 2815–19.
[36] Zhu X, Tang C, Wang P, et al. Saliency detection via affinity graph learning and weighted manifold ranking. Neurocomputing 2018;312(Oct):239–50.
[37] Zhou L, Yang Z, Zhou Z, Hu D. Salient region detection using diffusion process on a two-layer sparse graph. IEEE Trans Image Process 2017;26(Dec (12)):5882–94.
[38] Lie MMI, Borba GB, Vieira Neto H, Gamba HR. Joint upsampling of random color distance maps for fast salient region detection. Pattern Recognit Lett 2018;114(Oct):22–30.
[39] Perazzi F, Krähenbühl P, Pritch Y, Hornung A. Saliency filters: contrast based filtering for salient region detection. In: CVPR; 2012. p. 733–40.
[40] Fan D, Cheng M, Liu Y, Li T, Borji A. Structure-Measure: a new way to evaluate foreground maps. In: ICCV; 2017. p. 4548–57.
[41] Zhou D, Bousquet O, Lal TN, Weston J, Schölkopf B. Learning with local and global consistency. In: NIPS; 2004. p. 321–8.
[42] Zhou D, Weston J, Gretton A, Bousquet O, Schölkopf B. Ranking on data manifolds. In: NIPS; 2004. p. 169–76.
[43] Donoser M, Bischof H. Diffusion processes for retrieval revisited. In: CVPR; 2013. p. 1320–7.
[44] Achanta R, Shaji A, Smith K, et al. SLIC superpixels. EPFL; 2010. Tech. Rep. 149300.
[45] Zhang D, Song K, et al. Unified detection method of aluminium profile surface defects: common and rare defect categories. Opt Lasers Eng 2020;126(March):105936.
[46] Y. He, K. Song, et al., “An end-to-end steel surface defect detection approach via fusing multiple hierarchical features,” in IEEE Trans Instrum Meas. doi:10.1109/TIM.2019.2915404.