



Surface defect saliency of magnetic tile

Yibin Huang¹ · Congying Qiu² · Kui Yuan³

© Springer-Verlag GmbH Germany, part of Springer Nature 2018

Abstract

Computer vision builds a connection between image processing and industrials, bringing modern perception to the automated manufacture of magnetic tiles. In this article, we propose a real-time model called MCuePush U-Net, specifically designed for saliency detection of surface defect. Our model consists of three main components: MCue, U-Net and Push network. MCue generates three-channel resized inputs, including one MCue saliency image and two raw images; U-Net learns the most informative regions, and essentially it is a deep hierarchical structured convolutional network; Push network defines the specific location of predicted surface defects with bounding boxes, constructed by two fully connected layers and one output layer. We show that the model exceeds the state of the art in saliency detection of magnetic tiles, in which it both effectively and explicitly maps multiple surface defects from low-contrast images. The proposed model significantly reduces time cost of machinery from 0.5 s per image to 0.07 s and enhances detection accuracy for image-based defect examinations.

Keywords Saliency detection · Surface defect · Convolutional network

1 Introduction

Magnetic tiles provide constant magnetic potential as a key component of engines. Completing an automatic assembly line for magnetic tile manufacturer is a fundamental problem regarding its productivity, while still it is subjected to a few limitations. Surface defect detection is a core process of filtering unqualified products; however, in rare cases, the procedure can be finished automatically. It is recorded that as much as three quarters of workers are employed to examine product qualities in the magnetic tile factories in Zhejiang Province, China, the largest production base of magnetic tiles in the world. To relieve human labor, many image processing techniques have been proposed to attempt such inspection tasks; most of them are based on grayscale value and gradient edge of images, wavelet [1], curvelet [2] and shearlet [3,4] transformation. Deep neural network mod-

els have become dominant on the surface defect detection, promisingly achieving state-of-the-art performance on the classification tasks [5–7]. Overall, these explorations have proved computer vision a feasible solution on surface examinations.

To further explore potentials, a computer vision perspective readily requires our specific analysis on the characteristics of surface defect detection. In general, one magnetic tile contains 4–6 curved surfaces that cause images to be distorted. There have been several bottlenecks presented in the automatic damage detection for magnetic tiles, mostly due to the complexity of texture, the variety of defect shape, and the randomness of illumination conditions on magnetic tiles which likely develops into the noises of grayscale images. The target defects usually separate in curve surfaces without fixed patterns, such as blowhole, crack, break and fray, as shown in Fig. 1. These above-mentioned properties bring much randomness to image detection tasks; thus, we must overcome these inevitable challenges, especially when it comes to building computer vision models for robust results.

In the magnetic tile industry, image-based surface detection mostly concentrates on minimizing the inference of product texture. Embedding denoising techniques (e.g., wavelet [1], curvelet [2] and shearlet [3,4] transformation) can effectively help extract desirable features, but simultaneously lead to long running time. Most notably, the average

✉ Yibin Huang
huangyibin2014@ia.ac.cn

¹ Institute of Automation, Chinese Academy of Sciences, University of Chinese Academy of Sciences, No. 95, Zhongguancun East Road, Beijing 100190, China

² Civil Engineering and Engineering Mechanics Department, Columbia University, New York City, USA

³ Institute of Automation, Chinese Academy of Sciences, Beijing, China

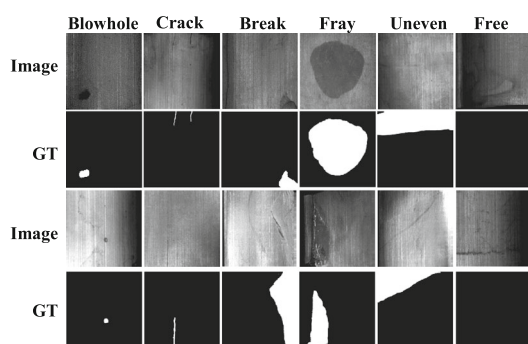


Fig. 1 Examples of <https://github.com/abin24/Magnetic-tile-defect-datasets> magnetic tile surface defects, labeled with pixel-level ground truth (GT)

processing time for each image is around 0.5 s, far from meeting the standard of real time. It is worth pointing out that the performance of an experienced maintenance worker still by far outweighs the machinery in terms of detection accuracy and efficiency. Replacing manual detection on magnetic tiles with automated surface inspection (ASI) [8] readily requires significant improvement on computer vision model performance.

Recent saliency detection models have been achieving promising performance on related tasks [9–11], exceeding in both feature extraction and running time. In fact, saliency detection has been introduced in computer vision for many years; hence, numerous state-of-the-art saliency detection models derived from this concept have been proposed in the past few decades. These models are designed to predict human fixations [12–14], driven by computer vision applications such as image segmentation [15], image abstraction [16], image manipulation [17], object retrieval [18], scene classification [9], 3D model simplification [19], or only focused at detecting the saliency objects [11,20,21]. In most cases, salient detection is merely distinct from semantic segmentation due to their high similarities in architectures and performance. Most distinguishably, the main purpose of saliency detection is to extract the most high-level features in the images. In terms of model selection for saliency detection, both detection accuracy and running time are two key aspects as we mention above. A considerable method is to standardize these models. Ali Borji, MingMing Cheng et al. made remarkable contributions to such model evaluations [22]. They qualitatively and quantitatively evaluate on 42 advancing saliency detection models mostly from top computer vision conferences such as CVPR, ICCV, ECCV and PAMI. They benchmark these models with respect to their algorithm performance using six datasets—MSRA10K [23], ECSSD [24], THUR15K [23], JuddDB [25], DUT-OMRON [26] and SED2 [27]. This significant work displays an intensive comparison among the 42 selected models not involved in the deep artificial neural network by using natural scene

image datasets. We have adopted the evaluation metrics and also compare some of the models in [22].

Practically, saliency detection models have an explosive and wide range of applications on various models of surface defect detection. For example, FT [28]-, MSS [29]- and ITTI [30]-like models are used in steel strip surface detection [31–33]; AC [34]-like models are used in weld inspection [35]; and PHOT [36]-like models are used in electronic chip inspection [37]. In this sense, it strongly demonstrates the potential of saliency detection models on surface defect inspection of magnetic tiles. However, in the above-mentioned benchmark commitment, the corresponding results may not serve as a universal solution for all saliency detection problems. The salient features of the six datasets are distinct from the case of surface defect detection on magnetic tiles, because most images of the six datasets are center surround, high contrast with image background, and big in size. In some cases, bokeh occurs in the scenes to emphasize target features. In fact, due to the distinguishable colors and positions of salient features, most models are able to perform detection tasks exceedingly well. Therefore, notably, the following factors should be taken into consideration if the benchmark is applied to surface defect detection:

1. Defects are randomly distributed and occasionally touch the boundaries of the images;
2. The colors of saliency features are very close to background; hence, grayscale images are used in industry. The images have lower background–foreground contrast and fewer available information;
3. Most defects are tiny and likely lie among stripe textures.

In principle, saliency detection on surface defects shares a lot of common theories on regular problems, while the differences of datasets make the detection task distinguish from conventional methodologies.

It is also worth noting that recent progress [13,38–43] on the saliency detection field is mostly deep learning based. For example, the caption-guided visual saliency, which is an extension of ConvNets, can explicitly learn spatial heatmaps from top-down inputs [38]; the adaptive-parameter multitask network can be trained end to end to produce a numerical representation of salient objects [40]. Most significantly, fully convolutional networks (FCNs) [44] have transferred the trend from classification into segmentation. This method produces a pixel-wise prediction for dense layers, in which it offers a fully convolutional perspective of existing networks and exceeds the state of the art. After FCNs, tremendous generative methodologies have been proposed to focus on the segmentation task, and U-Net [45] is among these cutting-edge models. In all, the revolutionary development of deep artificial neural network has provided our work with a number of options.

In the context, we explore if and how the saliency detection can be specifically applied to the surface detection of magnetic tiles. We propose an easy-operated and effective model that is structured by one U-Net embedded with the image preprocessing technique called MCue saliency herein and one Push network that defines the specific location of predicted surface defects with a bounding box. We also prove the feasibility of saliency detection on the surface defect of magnetic tiles.

This paper is organized as follows: In Sect. 2, we introduce a saliency toolbox for surface defect detection and the MCue saliency detection model customized for cracks and blowholes. In Sect. 3, we illustrate the architecture and training details of our proposed model. In Sect. 4, we compare the performance of saliency detection models with our proposed model and then discuss the saliency of different defects.

2 Multi-cue saliency

2.1 Toolbox

We aim to design a toolbox that is accessible, readable, realizable and transferable to the open-source community and to address the surface defect detection problem in industrial. There is tremendous computer vision research in top conferences bringing great inspirations to our work. The open-source community provides our toolbox with rich resources, and the chances to intensively test recent state-of-the-art algorithms and evaluate their performance on surface defect saliency. With excessive experiment records, we carefully evaluate, select and integrate a set of models; then, we produce an accessible public toolbox. A <https://github.com/abin24/Saliency-detection-toolbox> saliency detection toolbox is developed by embedding 14 algorithms which focus on the saliency detection of surface defects. For the ease of duplication, all the selected algorithms depend on OpenCV only. It is further noteworthy that the task-related models have low cost not only in time but also computational resources due to free from a training process. To simplify operations, these models do not contain hyperparameter setup, not necessarily to adjust parameter back and forth to get satisfactory predictions. These metrics provide a quick preview for defect type-orientated saliency, mostly referring to blowhole, crack and fray.

The 14 models in the toolbox are: ITTI [30] and BMS [14] based on visual-attention mechanism; FT [28], LC [46] and HC [47] based on global color rarity; AC [34] and MSS [29] based on local color rarity; SR [48], Rudinac [49] and PHOT [36] based on frequency domain; RC [47], SF [50] and GMR [26] based on region or superpixel contrast; and MBP [51] based on the fast minimum barrier distance (MBD) transform algorithm.

2.2 Saliency cue

We particularly get more insight into two main surface defects of magnetic tiles, blowhole and crack, and intensively analyze their characteristics on the grayscale raw images. We find five important cues that can be useful for constructing MCue saliency: (1) The reflective intensity of blowhole and crack is weaker than their surrounding environment; (2) Blowhole and crack have stronger corner and edge response, in most cases; (3) The gray values of defects are relatively significant in a local area; (4) The texture in image background is regular; (5) Human visual-attention mechanism can easily detect these two defects.

Darker cue The surface microgeometry of magnetic tiles changes once defects appear, causing the diffuse reflection of microsurfaces to change as well. Accordingly, defects such as blowhole and crack look darker than their surroundings. However, regular binarization still cannot precisely partition these defects because the illumination has not been normalized in the images. We hence use the adaptive binarization to calculate darker cue \mathcal{D} , in which it defines as:

$$\mathcal{D} = \begin{cases} 1 & \text{if } I_R - I > t \\ 0 & \text{otherwise} \end{cases} \quad (1)$$

I_R is mean filter blurred image of the raw image I in a $R * R$ blur window, and $t \geq 0$ is a constant threshold. When the grayscale value of a pixel is smaller than the mean of the $R * R$ local neighborhood, this pixel is considered darker than the other.

Strukturtensor cue The different diffuse reflection between the defect and non-defect area causes sharp edge or corner around the defect. The corner or edge responses of defects will be stronger than backgrounds. Harris [52] found that the strukturtensor (ST) (structure tensor) can be used for detecting corners and edges. Strukturtensor is the Hessian matrix of the image. Strukturtensor of an anchor pixel (x, y) is defined as:

$$M = \begin{bmatrix} G & F \\ F & H \end{bmatrix} = \begin{bmatrix} \frac{\partial^2 I}{\partial x^2} & \frac{\partial I}{\partial x} \cdot \frac{\partial I}{\partial y} \\ \frac{\partial I}{\partial x} \cdot \frac{\partial I}{\partial y} & \frac{\partial^2 I}{\partial y^2} \end{bmatrix} \quad (2)$$

Let λ_1 and λ_2 be the eigenvalues of M , then we get:

$$\begin{cases} \lambda_1 = \frac{G+H+\sqrt{(G-H)^2+4F^2}}{2} \\ \lambda_2 = \frac{G+H-\sqrt{(G-H)^2+4F^2}}{2} \end{cases} \quad (3)$$

Harris [52] proves that the edge response will occur when one eigenvalue is large, while the other one is small, and the corner response happens if and only if both two eigenvalues are

large. So let $A = (\lambda_1 - \lambda_2)^2$ represents the edge response of the image, and $B = |\lambda_1 + \lambda_2|$ represents the corner response of an anchor pixel. with Eq. (3), and then, we get:

$$\begin{cases} A = (G - H)^2 + F^2 \\ B = G + H \end{cases} \quad (4)$$

Then, the definition of strukturtensor of the image is:

$$S_{ST} = \frac{\mathcal{N}(SalA) + \mathcal{N}(SalB)}{2} \quad (5)$$

where $SalA$ and $SalB$ are computed with Eq. 4, which represent the edge and the corner saliency map of an input image, and $\mathcal{N}(\cdot)$ represents the normalized operation of the saliency map.

2.3 Cue fusion

In the third cue, we point out that the gray level of blowholes and cracks is distinguishable from the surroundings, in which the defect color easily grabs attention because of its rarity. The gray level of blowholes and cracks that is locally rare may not be rare in the entire image. Capturing color rarity helps precisely localize saliency features, consequently diminishing the computational complexity for the network. AC model is selected to perform this specific task. It computes the target pixel quality and measures the Euclidean distance between a selected pixel value and average pixel values of multiple neighborhoods with different sizes. The color rarity of local regions is thus obtained.

We also mention that according to the fourth cue we list, the distribution of background texture follows fixed patterns. In most cases, the texture is approximately parallel and heavy, producing a lot of noise to the prediction. PHOT algorithm has been integrated into the toolbox for the purpose of eliminating such the interference of background textures. BMS model is to detect saliency objects by simulating the human visual-attention system.

Based on the above-mentioned cues, we propose two saliency cues called $MCue$ and $MCue2$, respectively. They are constructed as:

$$MCue = (\mathcal{D}\mathcal{W}_{\mathcal{D}} + 1.0) \times (S_{BMS}\mathcal{W}_{BMS} + S_{AC} + S_{ST} + S_{PHOT}\mathcal{W}_{PHOT})/m \quad (6)$$

and

$$MCue2 = S_{BMS} \times (\mathcal{D}\mathcal{W}_{\mathcal{D}} + 1.0) \times (S_{AC} + S_{ST} + S_{PHOT}\mathcal{W}_{PHOT})/n \quad (7)$$

where S_{BMS} , S_{AC} , S_{ST} , S_{PHOT} and \mathcal{D} , respectively, represent the saliency map of BMS, AC, Strukturtensor and darker

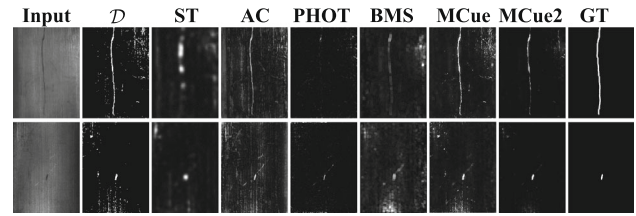


Fig. 2 Saliency map comparison. $MCue2$ has better fusion than $MCue$

cue. $\mathcal{W}_{\mathcal{D}}$, \mathcal{W}_{BMS} , \mathcal{W}_{PHOT} are the weights of \mathcal{D} , BMS, PHOT saliency map, and m, n are the normalization constants. Figure 2 shows the results of multi-cue fusion.

Image pixel-wise addition interactively complements the saliency regions from each operating image; those regions are thus strengthened. At the same time, the image multiplication only strengthens the areas with high saliency probability in both images, and areas with low saliency probability in either image will be weakened.

We implement pixel-wise additions and multiplications on image fusion, in which $\mathcal{W}_{\mathcal{D}} = \mathcal{W}_{BMS} = \mathcal{W}_{PHOT} = 3$, $m = 4$ and $n = 5$. The results are shown in Fig. 2. In the test on ST, PHOT and AC, most of high probability density regions are correctly distinguished as surface defects, but it still cannot map the entire defect shape. Therefore, we use the image addition to complement these saliency maps. PHOT saliency can precisely identify the defects according to the high probability density of these regions, so we assign \mathcal{W}_{PHOT} with a high value. Simultaneously, the direct multiplication likely leads to false detection because of the strong restriction to darker cue \mathcal{D} . We conduct image multiplication with a weight $\mathcal{W}_{\mathcal{D}}$ with additional 1.0 to loosen the constraint of the darker cue. Figure 2 shows that BMS is able to identify all the defects, while non-defect regions have high values as well. Image multiplication diminishes these values that are erroneously detected as defects. This is why $MCue2$ produces more accurate saliency maps than $MCue$. More comparison details are displayed in Sect. 4.

3 U-Net-based saliency

Deep learning dominantly drives advances on saliency detection because of the state-of-the-art performance. In fact, most architectures of cutting-edge saliency detection models [38–41] are generated from fully convolutional network [44] (FCN). The architecture of FCNs is an encoder–decoder structure because it takes the input of arbitrary size and produces correspondingly sized output. Multiple convolutional layers are hierarchically structured, and each convolutional layer is followed by a pooling layer. Desirable high-level semantic features are captured when the convolutional layer is down-sampled by the pooling layer. However, the pooling

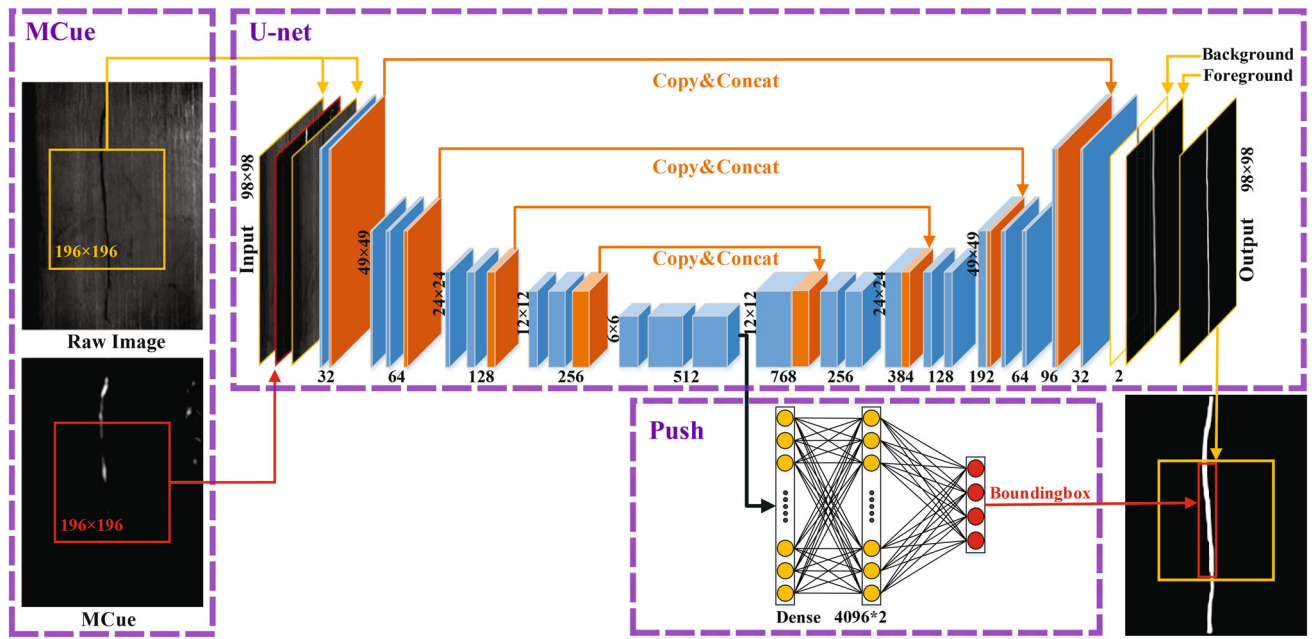


Fig. 3 An architecture overview of MCuePushU with three main components: MCue, U-Net and Push network. Two MCue saliency maps are cropped into 196*196, together with the corresponding 196*196 raw images, resized as new 98*98*3 inputs. U-Net pixel to pixel learns

features. The convolutional layer of U-Net is directly copied and arbitrarily cropped to attain a same size up-sampling layer. Push network has two fully connected layers with the size of 4096*4096 and an output layer. It highlights the most informative regions with a bounding box

process inevitably leads to information loss with respect to feature localization.

U-Net [45] is a significant modification and an extension of FCNs, originally designed for biomedical cell image segmentation. This elegant network not only avoids key information loss but also enables the model to be learned pixel to pixel and end to end. The encoder and the decoder of U-Net are distinct from those of FCNs, where a convolutional layer of U-Net is directly copied and arbitrarily cropped to attain a same size up-sampling layer. Skipping the fixed hierarchy connections helps repair detail information lost in the pooling process. Such architecture has been applied in many advancing saliency detection models for more accurate saliency map.

3.1 Proposed methodology

To identify surface defects of magnetic tiles, we propose a saliency detection model MCuePush U-Net, which is illustrated in Fig. 3. The input layer is a 98*98*3 tensor including one MCue2 saliency image and two raw images. 196*196 region of interest (ROI) is cropped from the MCue2 saliency map and raw images and resized to 98*98 as the network input. Practically, smaller networks run faster and need less overhead, so an input size with 98*98 is used. The reason of using three-channel input is twofold: first, it is easier to store the training images on the disk through the storage for-

mat of the three-channel image; second, we have also test two-channel (1-channel raw image with 1-channel MCue2) input, and the two-channel input models are severely overfitting. Thus, we decide to use the three-channel image as the input of the model.

Softmax is selected as the activation function to present a two-categorical distribution in the range [0, 1], which produces scores, respectively, corresponding to the foreground and background. Thus, the final output has two independent channels in terms of two-dimensional scores. In the MCuePushU, we particularly focus on the saliency map/foreground, where its predicted probability is:

$$Pred = \frac{e^f}{e^f + e^b} \quad (8)$$

where $Pred$ is the score/predicted probability of foreground and f and b is, respectively, the foreground and background of output images.

To separate the foreground and background from the above-mentioned scores, we use the Softmax classifier with a cross-entropy loss, which has the form:

$$\mathcal{L} = - \sum_p (L_p \log(Pred_p) + (1 - L_p) \log(1 - Pred_p)) \quad (9)$$

where we use L_p to mean the ground-truth label of pixel p and use $Pred_p$ to represent the predicted value of pixel p . The word “Push” in MCuePushU refers to an independent structure that is distinct from U-Net, where we herein call it Push network. In the Push network, two layers with the size of 4096×4096 are fully connected together, followed by an output layer, defining the specific location of predicted surface defects with a bounding box.

We apply the Euclidean algorithm to compute the loss. Given the ground-truth central coordinates (x_L, y_L) , width w_L and height h_L of predicted defects, the loss function is formulated as:

$$\begin{aligned} \mathcal{L}_{Push} &= ||e_x, e_y, e_w, e_h|| \\ &= (x_L - x_{Pred})^2 + (y_L - y_{Pred})^2 \\ &\quad + (w_L - w_{Pred})^2 + (h_L - h_{Pred})^2 \end{aligned} \quad (10)$$

The label (x_L, y_L, w_L, h_L) includes the location information of bounding box, which is generated from the ground truth of raw images. $x_{Pred}, y_{Pred}, w_{Pred}, h_{Pred}$ represent the central coordinates, width and height of predicted bounding box.

Push network can only deal with those images which have one defect, we do not update the network when the training images have more than one or no defect regions. Besides, in most case, defects will appear no more than once in a 196×196 ROI. Push network forces the most down-sampled layer to learn the highest-level image semantics in MCuePushU with a bounding box. It retrieves information including the location and the size of surface defects. In this designed architecture, the encoder gradually embeds the most high-level information, while the decoder dedicatedly repairs key information loss and improves prediction accuracy. Push network plays a crucial role to help U-Net approach to desirable saliency.

3.2 Implementation details

In the training phase, ROI is cropped from training images and their corresponding MCue2 saliency maps. Only the ROIs with the defect regions are used for training in consideration of the severe unbalance of the defect pixels and background pixels. In the testing phase, a size of 196×196 sliding window is used on the test images and their MCue2 saliency maps, ROI is down-sampled as a 98×98 input, and an output image is up-sampled back to 196×196 . Full-resolution saliency maps can be obtained by scanning the whole test images, and overlap parts will be assigned by the pixel-wise sum. The Push network can be abandoned in the testing phase, because the bounding box is not the ultimate desired result; besides, it also releases a considerable computation workload of the fully connected network.

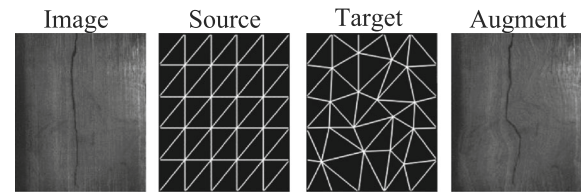


Fig. 4 MLS deformation data augmentation

Labeling pixel-level ground truth, most notably for grayscale images of magnetic tiles, requires tremendously heavy workload. It hence seems implausible to attain a great deal of such data, causing the training process more challenging. This limitation leads a consideration of strong data augmentation to serve our saliency network.

There have existed enormous techniques for data augmentation, and the most commonly seen methods include image rotation, flip, crop and transpose. It is worth noting that data augmentation on single small dataset possibly causes overfitting. To prevent the model from overfitting, we introduce a distinct data augmentation, which is an image deformation method using moving least square (MLS) [53]. As shown in Fig. 4, with 5×5 grids and Delaunay triangulation, the image is divided into 50 even triangular pieces, and the apexes of the triangles will be randomly shifted 0–40 pixels to build a target Delaunay mesh. The affine transform matrix of each triangle from the source Delaunay mesh to the target Delaunay mesh is computed with the MLS, and then, the image will be piecewise deformed with these 50 matrixes. Smooth local affine deformation images can be obtained by this image augmentation method. In the process, raw images, together with the corresponding ground-truth labels and MCue saliency maps, are used for augmenting. Each training image is augmented into 12 images using MLS; then, 196×196 ROIs are cropped, 30 times data size of themselves with regular augmentations, including rotation, translation, zoom and shear transform. We thus attain around 31,000 images along with ground-truth labels for network training.

Stochastic gradient descent (SGD) with momentum has been used as the optimization algorithm in U-Net, where we set momentum term to be 0.9 and learning rate to be 0.0001, and mini-batch size is set as 8 due to the limited graphics memory. Multiple losses are involved in the network, because MCuePushU is trained as a whole, simultaneously minimizing the loss functions of U-Net and Push network. One single epoch consists of 400 iterations. However, during the optimizing process, the loss function of Push network converges less significantly than that of U-Net. On account of that both loss functions of U-Net and Push network are optimized in the first 250 iterations, while only the Push network continues updating in the following 150 iterations.

As we allude, saliency defect detection is essentially a binary classification problem, in which images are separated

into two classes: foreground and background. It is worthwhile to mention that most cases the saliency maps are very tiny; therefore, pixels of the foreground class are far less than of the rest one. When the foreground pixels are mistakenly categorized, loss values may not be updated any longer because the decrement is too small. Thus, the optimization is stuck in wrong local minimum. It may occur that output images ultimately turn into all black. In such cases, users should restart the training process to re-initialize parameters. The runtime for 42,000 iterations is 10 h on the laptop equipped with Intel i7 7th GEN CPU, 8 G RAM and one GTX 1050ti Graphics.

4 Experiment

4.1 Dataset and evaluation metrics

Datasets A total of 1344 images are photographed, with the ROI of magnetic tile cropped and then classified into six datasets in terms of their defect types. The six datasets are named as: *Blowhole*, *Crack*, *Fray*, *Break*, *Uneven* (caused by grinding process) and *Free* (no defects); each has pixel-level labels. To simulate the manufacturing process in real assembly line, for one given magnetic tile we collect images under multiple illumination conditions.

Blowhole and *Crack* impact the quality of magnetic tiles the most: There are plenty of common features between the two defects; the color of them is darker than the surroundings. *Uneven* has the utmost difficulty to detect due to the fact that its color and texture are highly semblable to the background. *Fray* has similar color and texture as dark surface dirt, shown in the 6th row of Fig. 1. However, the dirt does not affect the functionality of magnetic tile. These defects, especially *Uneven* and *Fray*, bring unexpected challenges even for human beings.

According to the above-mentioned characteristics of *Blowhole* and *Crack*, we merge the images of these two defects into one dataset, namely *Blowhole&Crack*. We also assemble all the datasets into another one called *All* for the purpose of model evaluation. 50% randomly selected images of the datasets are used for training the models, and the rest for test.

Evaluation Metrics We compare the saliency map with corresponding ground truth to quantitatively evaluate the saliency models. Notably, one crucial process of evaluation is to transform saliency binarization into binarization mask, using either fixed or adaptive threshold methods. The fixed threshold method uses a series of uniform distributed threshold, while the adaptive threshold T_{Adv} is formulated by:

$$T_{Adv} = \frac{2}{W \times H} \sum_{x=1}^W \sum_{y=1}^H S(x, y) \quad (11)$$

where W and H are the width and height of saliency map S , respectively.

To get the measure of model performance, we use precision–recall (**PR**) curve and receiver operating characteristic (**ROC**) curve as two important factors. PR curve turns saliency binarization into binarization mask with a set of fixed thresholds, compares the mask with ground truth and records the precision and recall rate. The form of precision and recall rate with respect to mask M and ground truth G is:

$$\text{Precision} = \frac{|M \cap G|}{|M|} \quad \text{Recall} = \frac{|M \cap G|}{|G|} \quad (12)$$

F-measure puts precision and recall into the same measuring metric, which is defined as:

$$F_\beta = \frac{(1 + \beta^2) \text{Precision} \times \text{Recall}}{\beta^2 \text{Precision} + \text{Recall}} \quad (13)$$

where we set $\beta^2 = 0.3$ to pay more attention to precision. We also record the maximum F_β^{Max} from PR curve and compute F_β^{Adp} using adaptive binarization.

ROC curve is also widely used to evaluate saliency models, which records true-positive rate (TPR) and false-positive rate (FPR). The form of TPR and FPR follows:

$$TPR = \frac{|M \cap G|}{|G|} \quad FPR = \frac{|M \cap \bar{G}|}{|\bar{G}|} \quad (14)$$

Mean absolute error (**MAE**) is defined by (15). It summarizes model performance in ways that disregard the direction of over- or under-prediction, in which it punishes the model that mistakenly identifies background as foreground. However, there exists a drawback in this measurement. When image background is much larger than the foreground, the model will classify a whole image into the background. In such case, MAE still returns positive feedback. At the same time, the pixel number of background far exceeds foreground, causing PR curve more informative than ROC. Therefore, F_β contains more information than area under ROC (AUC) that represents the area below ROC curve, because F_β is related to PR while AUC to ROC.

$$MAE = \frac{2}{W \times H} \sum_{x=1}^W \sum_{y=1}^H |\bar{S}(x, y) - \bar{G}(x, y)| \quad (15)$$

4.2 Model improvement

MCue model Our focus lies on the comparison between MCue2 and MCue. The experiment demonstrates that in most aspects MCue2 performs better than MCue, though there is only a few difference between their fusion methods. For example, compared with MCue, Fig. 2 shows that

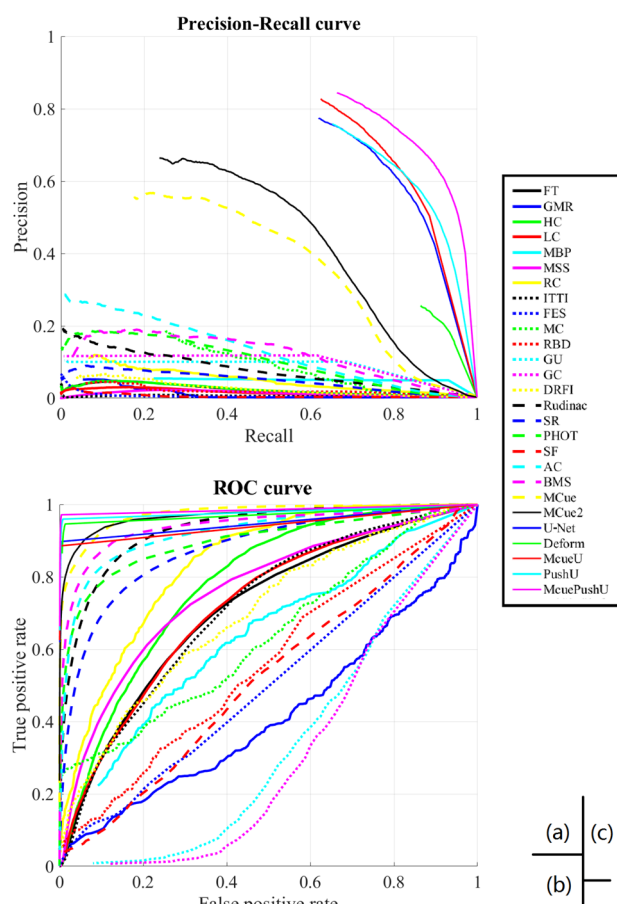


Fig. 5 Model comparison of PR curve (a) and ROC curve (b) on the *Blowhole&Crack* dataset

MCue2 is closer to ground truth; Fig. 5 reveals that both the PR and ROC curves of MCue2 have dominant performance; Table 1 also displays that MCue2 get higher scores in most terms. Conspicuously, MCue2 outperforms all the other models except for deep neural network (U-Net) on the *Blowhole&Crack* dataset, which is shown as the first four columns in Table 1.

MCuePushU We use MCuePush U-Net, namely MCuePushU, to compare with the original U-Net and deformable U-Net.

Deformable U-Net [54] is built upon U-Net architecture, using deformable convolutional kernels [55] that adaptively learn spatial sampling locations with different sizes. The model does not need data augmentation because it learns spatial invariance features during the training process [54]. Deformable U-Net attains better results than standard U-Net in the biological image segmentation task.

To testify the availability of our proposed components based on MCue and U-Net, we conduct the experiments, respectively, on Push, MCue U-Net (herein called MCueU) and Push U-Net (herein called PushU). The hyperparameters

are set into the same in the experiment comparison, including input size, kernel amount, batch size, learning rate and iteration number. Data augmentation is still applied to increase the dataset variance.

In the analysis of PR and ROC curves, Fig. 5 evidently reveals that MCuePushU is dominant over all the other models; the performance of PushU and MCueU is both slightly better than U-Net. In Table 1, the scores of MCuePushU drive exceeding advances in contrast to the rest, while the deformable U-Net gets very poor results on account of over-fitting, as shown in Fig. 6. Notably, this network has wrongly classified a large amount of background pixel into the foreground class. In addition, the processing time of deformable U-Net is around 0.7 s per image, 10 times longer than MCuePushU that only needs 0.07 s per image. MCuePushU is also much faster than the average speed 0.5 s of wavelet [1], curvelet [2] and shearlet [3,4].

4.3 Model performance on defect datasets

In this section, we focus on exploring the saliency of the surface defect by intensive and comprehensive comparisons with respect to MCue model. We compare the performance of MCue model with 14 models in the toolbox. At the same time, we also investigate RBD [56], FES [57], MC [58], GC [16], GU [16] and DRFI [59]. Notwithstanding their good saliency performance, the open-source code of these models may not be readily duplicated because they are not based on C++. Additionally, GC and GU contain a great number of contents that are irrelevant to saliency, making the project transferring process complicated. For the ease of transfer, most cases we select C++-based algorithms; these models hence have not been integrated into the toolbox.

Figure 5 shows that, in the analysis of *Blowhole&Crack* dataset, the PR and ROC curves of MCue outweigh all models that are not deep learning based. BMS, AC, PHOT, SR and Rudinac perform secondly only to U-Net. Notably, MCue dominates over all the other models in ROC. Figure 7 displays the saliency detection results. FT, AC, LC, HC and MSS models satisfactory saliency performances for the reason that they take local or global saliencies into consideration. Some models, such as MC, RBD, GMR, RC and SF, are region and superpixel based, in which they usually cannot identify the difference between blowhole and crack. Frequency domain analysis-based models such as SR, Rudinac and PHOT are good at blowhole detection, even for the smallest ones, while these models are usually not capable of detecting entire crack shapes.

The saliency results on the *Fray* dataset are shown in the last two rows of Fig. 7. The shape of fray is significantly larger than of blowhole and crack. When fray appears in the middle without hitting the image boundaries (as shown in the fourth row), the saliency map of the region- and superpixel-based

Table 1 Model performance comparison on different datasets: Blowhole&Crack, Fray and All

Dataset	<i>Blowhole&Crack</i>				<i>Fray</i>				<i>All</i>			
Model	AUC	MAE	F_{β}^{Max}	F_{β}^{Adp}	AUC	MAE	F_{β}^{Max}	F_{β}^{Adp}	AUC	MAE	F_{β}^{Max}	F_{β}^{Adp}
FT	0.710	0.073	0.056	0.162	0.575	0.218	0.216	0.160	0.899	0.093	0.039	0.058
GMR	0.420	0.675	0.058	0.095	0.595	0.494	<i>0.448</i>	0.025	0.815	0.662	0.035	0.018
HC	0.780	0.119	0.057	0.223	0.555	0.252	0.212	0.160	0.909	0.137	0.041	0.067
LC	0.717	0.056	0.038	0.209	0.568	0.211	0.215	0.161	0.898	0.079	0.041	0.064
MBP	0.625	0.435	0.067	0.202	<i>0.716</i>	0.279	<i>0.418</i>	0.142	0.797	0.439	0.082	0.047
MSS	0.760	0.019	0.029	0.279	0.527	0.180	0.248	0.160	0.894	<i>0.041</i>	0.042	0.077
RC	0.837	0.174	0.112	0.241	<i>0.720</i>	0.273	0.315	0.158	0.921	0.182	0.050	0.069
ITTI	0.712	0.339	0.012	0.205	<i>0.660</i>	0.356	0.242	0.175	0.915	0.324	0.042	0.065
FES	0.504	0.042	0.031	0.125	0.627	<i>0.175</i>	0.388	0.103	0.840	0.065	0.032	0.030
MC	0.633	0.361	0.204	0.267	0.630	0.343	<i>0.460</i>	0.154	0.843	0.425	0.052	0.052
RBD	0.558	0.130	0.053	0.142	0.577	0.221	0.327	0.090	0.855	0.144	0.045	0.044
GU	0.338	0.446	0.126	0.016	0.351	0.501	0.242	0.139	0.744	0.491	0.095	0.022
GC	0.313	0.446	0.146	0.001	0.364	0.464	0.269	0.112	0.712	0.488	<i>0.098</i>	0.018
DRFI	0.695	0.340	0.074	0.274	0.541	0.351	0.256	0.110	0.878	0.374	0.046	0.064
Rudinac	0.930	0.036	0.140	0.004	0.606	0.187	0.254	<i>0.212</i>	<i>0.934</i>	0.057	0.063	0.076
SR	0.877	0.047	0.090	0.296	0.478	0.190	0.212	<i>0.178</i>	0.919	0.065	0.048	<i>0.080</i>
PHOT	0.898	0.008	0.185	0.268	0.597	<i>0.173</i>	0.212	0.158	0.926	<i>0.030</i>	0.062	0.074
SF	0.520	0.143	0.029	0.157	0.618	0.251	0.322	0.130	0.849	0.174	0.038	0.046
AC	0.923	0.098	0.223	0.261	0.643	0.225	0.226	0.174	<i>0.932</i>	0.122	0.063	0.073
BMS	0.938	0.071	0.193	0.419	0.510	0.213	0.212	0.156	0.925	0.097	0.069	<i>0.093</i>
MCue	<i>0.974</i>	0.032	0.490	0.331	0.628	0.189	0.212	0.129	0.942	0.058	<i>0.100</i>	0.074
MCue2	<i>0.968</i>	<i>0.007</i>	<i>0.560</i>	0.268	0.576	<i>0.174</i>	0.212	<i>0.185</i>	<i>0.935</i>	<i>0.032</i>	<i>0.113</i>	<i>0.081</i>
U-Net	0.948	0.002	<i>0.732</i>	<i>0.531</i>	0.880	0.064	0.793	0.250	0.956	0.027	0.174	0.113
Deform	<i>0.967</i>	0.009	0.306	<i>0.495</i>	—	—	—	—	—	—	—	—
MCueU	0.942	0.002	0.77	0.535	—	—	—	—	—	—	—	—
PushU	0.978	<i>0.003</i>	<i>0.731</i>	<i>0.527</i>	0.923	0.056	0.808	0.270	0.966	0.022	0.164	0.114
MCuePushU	0.985	0.002	0.795	0.549	—	—	—	—	—	—	—	—

The highest score of each index is marked in bold, the second in bold italic and the rest of top 5 in italic

models (GMR, MC, RBD, RC and SF) is satisfactory, while the results turn into blurred and ambiguous if fray stretches into image boundaries (as shown in the third row). It is worth noting that MPB, a model that based on minimum barrier distance, generates very similar results with those models that are region and superpixel based. In Table 1, statics based on the *Fray* dataset is consistent with the above-mentioned conclusions, in which GMR, MBP, RC and MC get some top 5 scores. We also notice that center bias has introduced a lot of randomnesses because surface defects may appear in any location. In the worst case, the output of FES model only contains center bias without detecting desirable objects.

In Table 1, the last four columns present the scores on All dataset. The statics reveals that DRFI, RBD, MC, GMR cannot generate good results in our case, which is contrary to the study of Borji' [22] where these models are the top ones on natural image dataset. A reasonable interpretation

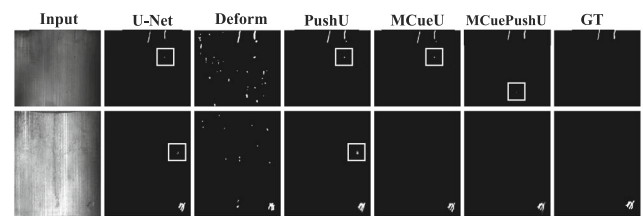


Fig. 6 The improvement of the MCuePushU. White bounding boxes in the images are false alarms, and MCuePushU has fewer false alarms

is saliency models generated from natural images which are not applicable for industrial production.

MCue is particularly designed on basis of the *Blowhole&Crack* dataset, so we do not test MCuePushU on other datasets because it has integrated MCue. Another deep neural network model is PushU, which is developed on *Fray* and *All* dataset. It places emphasis on the network atten-

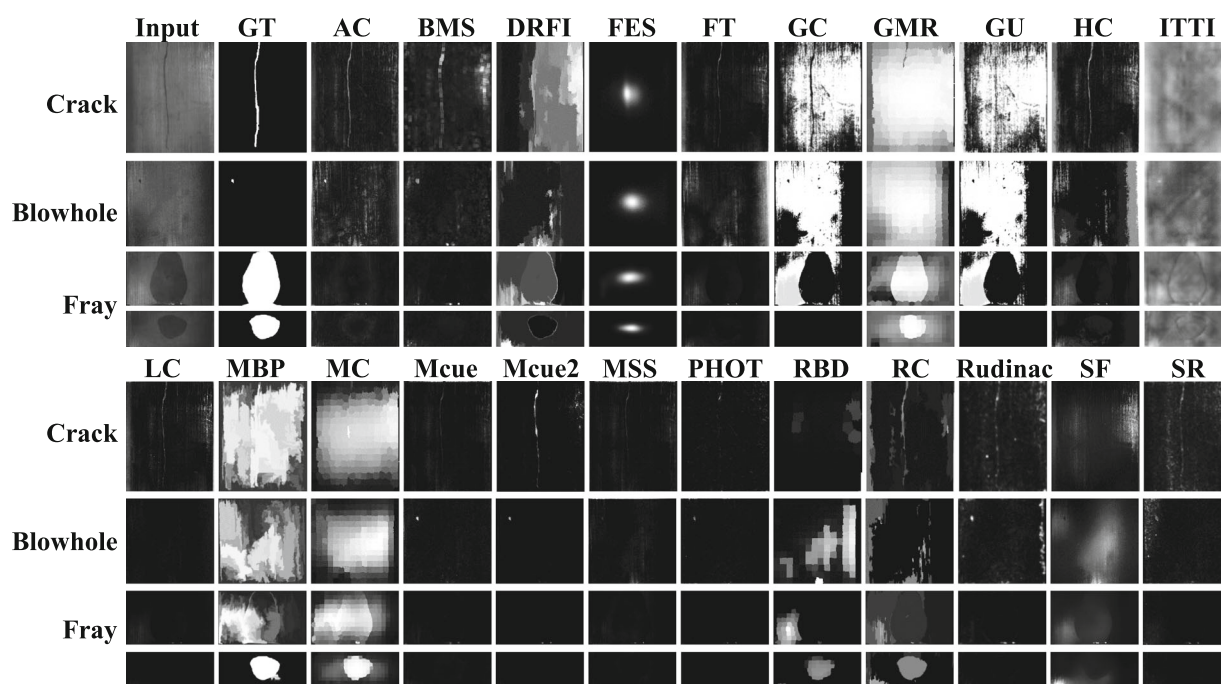


Fig. 7 Model performance on different defect datasets

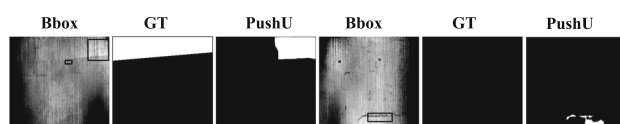


Fig. 8 False detection of PushU on *All* dataset, especially for uneven surface caused by grinding process, and dark surface dirt

tion with black bounding boxes. The models performance is superior to U-Net in all aspects. However, on the all dataset PushU still fails to detect some defects as shown in Fig. 8. For example, the dirt on the magnetic tile is actually non-defect, while PushU cannot correctly distinguish it. In fact, these undetectable defects only take a few proportions of magnetic tiles, so such influence can be ignored. Practically, our models have promising application in the magnetic tile industry.

5 Conclusion

We explore the potential of image saliency in the industrial application: magnetic tile defect detection. The *Blowhole&Crack* particularly draws our attention due to its crucial impact on the magnetic tile manufacturing. We propose a customized model named MCuePushU; this model summarizes a set of dominant cues, then fuses them into the deep neural network U-Net through image arithmetic and finally embeds a Push network to highlight the predicted defects

with bounding boxes. Experiments demonstrate that MCuePushU achieves state-of-the-art saliency performance as well as meets the demand of real-time inspection process, exceedingly outperforming all the other models tested in this article.

The experiment also corroborates the assumption that the performance of models varies in terms of datasets. BMS, AC, PHOT, SR and Rudinac models exceed other models on the *Blowhole&Crack* dataset; frequency domain analysis-based models show strength in tiny defect detection; superpixel-based models can accurately map large center-surround features. These facts can provide useful guidance for automated production of magnetic tiles.

There still exist some defects that our models fail to detect. In future, we will focus on the improvement of algorithms to further satisfy comprehensive defect detection.

Acknowledgements This study is supported by National Natural Science Foundation of China (Grant No. 61421004).

Compliance with ethical standards

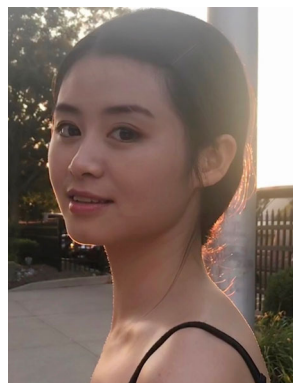
Conflict of interest Authors Yibin Huang, Congying Qiu, Maria Feng and Kui Yuan declare that they have no conflict of interest.

References

1. Yang, C., Liu, P., Yin, G., Jiang, H., Li, X.: Defect detection in magnetic tile images based on stationary wavelet transform. *NDT & E Int.* **83**, 78–87 (2016)

2. Li, X., Jiang, H., Yin, G.: Detection of surface crack defects on ferrite magnetic tile. *NDT & E Int.* **62**, 6–13 (2014)
3. Xie, L., Lin, L., Yin, M., Meng, L., Yin, G.: A novel surface defect inspection algorithm for magnetic tile. *Appl. Surf. Sci.* **375**, 118–126 (2016)
4. Yang, C., Liu, P., Yin, G., Wang, L.: Crack detection in magnetic tile images using nonsubsampling shearlet transform and envelope gray level gradient. *Opt. Laser Technol.* **90**, 7–17 (2017)
5. Cha, Y.-J., Choi, W., Büyükoztürk, O.: Deep learning-based crack damage detection using convolutional neural networks. *Comput. Aided Civ. Infrastruct. Eng.* **32**(5), 361–378 (2017)
6. Masci, J., Meier, U., Ciresan, D., Schmidhuber, J., Fricout, G.: Steel defect classification with max-pooling convolutional neural networks. In: *The 2012 International Joint Conference on Neural Networks (IJCNN)*, pp. 1–6. IEEE (2012)
7. Soukup, D., Huber-Mörk, R.: Convolutional neural networks for steel surface defect detection from photometric stereo images. In: *International Symposium on Visual Computing*, pp. 668–677. Springer (2014)
8. Ren, R., Hung, T., Tan, K.C.: A generic deep-learning-based approach for automated surface inspection. *IEEE Trans. Cybern.* **48**(3), 929–940 (2018)
9. Yang, C., Pu, J., Dong, Y., Xie, G.S., Si, Y., Liu, Z.: Scene classification-oriented saliency detection via the modularized prescription. *Vis. Comput.* **1**, 1–16 (2018)
10. Yang, Z., Xiong, H.: Computing object-based saliency via locality-constrained linear coding and conditional random fields. *Vis. Comput.* **33**(11), 1403–1413 (2017)
11. Zhou, X., Wang, Y., Zhu, Q., Xiao, C., Xiao, L.: Ssg: superpixel segmentation and grabcut-based salient object segmentation. *Vis. Comput.* **11**, 1–14 (2018)
12. Gorji, S., Clark, J.J.: Attentional push: a deep convolutional network for augmenting image saliency with shared attention modeling in social scenes. In: *IEEE Conference on Computer Vision and Pattern Recognition*, pp. 3472–3481, (2017)
13. Ramanishka, V., Das, A., Zhang, J., Saenko, K.: Top-down visual saliency guided by captions. In: *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*, vol. 1, p. 7 (2017)
14. Zhang, J., Sclaroff, S.: Saliency detection: a boolean map approach. In: *IEEE International Conference on Computer Vision*, pp. 153–160 (2013)
15. Donoser, M., Urschler, M., Hirzer, M., Bischof, H.: Saliency driven total variation segmentation. In: *IEEE International Conference on Computer Vision*, pp. 817–824 (2009)
16. Cheng, M.M., Warrell, J., Lin, W.Y., Zheng, S., Vineet, V., Crook, N.: Efficient salient region detection with soft image abstraction. In: *IEEE International Conference on Computer Vision*, pp. 1529–1536 (2013)
17. Ran, M., Zelnik-Manor, L., Tal, A.: Saliency for image manipulation. *Vis. Comput.* **29**(5), 381–392 (2013)
18. Gao, Y., Wang, M., Tao, D., Ji, R., Dai, Q.: 3-d object retrieval and recognition with hypergraph analysis. *IEEE Trans. Image Process.* **21**(9), 4290–4303 (2012)
19. Yang, B., Li, F.W., Wang, X., Mingliang, X., Liang, X., Jiang, Z., Jiang, Y.: Visual saliency guided textured model simplification. *Vis. Comput.* **32**(11), 1415–1432 (2016)
20. Tang, Y., Tong, R., Tang, M., Zhang, Y.: Depth incorporating with color improves salient object detection. *Vis. Comput.* **32**(1), 111–121 (2016)
21. Kapoor, A., Biswas, K.K., Hanmandlu, M.: An evolutionary learning based fuzzy theoretic approach for salient object detection. *Vis. Comput.* **33**(5), 1–21 (2016)
22. Borji, A., Cheng, M.M., Jiang, H., Li, J.: Salient object detection: a benchmark. *IEEE Trans. Image Process.* **24**(12), 5706–5722 (2015)
23. Thur15000. <http://mmcheng.net/gsal/>. Accessed 26 Mar 2018
24. Yan, Q., Xu, L., Shi, J., Jia, J.: Hierarchical saliency detection. In: *Computer Vision and Pattern Recognition*, pp. 1155–1162 (2013)
25. Borji, A.: What is a salient object? a dataset and a baseline model for salient object detection. *IEEE Trans. Image Process. A Publ. IEEE Signal Process. Soc.* **24**(2), 742 (2015)
26. Yang, C., Zhang, L., Lu, H., Xiang, R., Yang, M.H.: Saliency detection via graph-based manifold ranking. In: *IEEE Conference on Computer Vision and Pattern Recognition*, pp. 3166–3173 (2013)
27. Alpert, S., Galun, M., Brandt, A., Basri, R.: Image segmentation by probabilistic bottom-up aggregation and cue integration. *IEEE Trans. Pattern Anal. Mach. Intell.* **34**(2), 315–26 (2012)
28. Achanta, R., Hemami, S., Estrada, F., Susstrunk, S.: Frequency-tuned salient region detection. In: *IEEE Conference on Computer Vision and Pattern Recognition. CVPR 2009*. pp. 1597–1604 (2009)
29. Achanta, R., Ssstrunk, S.: Saliency detection using maximum symmetric surround. In: *IEEE International Conference on Image Processing*, pp. 2653–2656 (2010)
30. Itti, L., Koch, C., Niebur, E.: A model of saliency-based visual attention for rapid scene analysis. *IEEE Trans.* **20**(11), 1254–1259 (1998)
31. Song, K.C., Hu, S.P., Yan, Y.H., Li, J.: Surface defect detection method using saliency linear scanning morphology for silicon steel strip under oil pollution interference. *ISIJ Int.* **54**(11), 2598–2607 (2014)
32. Song, K., Yan, Y.: Micro surface defect detection method for silicon steel strip based on saliency convex active contour model. *Math. Probl. Eng.* **2013**(8), 1–13 (2013)
33. Guan, S.: Strip steel defect detection based on saliency map construction using gaussian pyramid decomposition. *Trans. Iron Steel Inst. Jpn.* **55**(9), 1950–1955 (2015)
34. Achanta, R., Estrada, F., Wils, P., Ssstrunk, S.: Salient region detection and segmentation. In: *International Conference on Computer Vision Systems*, pp. 66–75 (2008)
35. Gharsallah, M.B., Braiek, E.B.: Weld inspection based on radiography image segmentation with level set active contour guided off-center saliency map. *Adv. Mater. Sci. Eng.* **2015**(11), 1–10 (2015)
36. Aiger, D., Talbot, H.: The phase only transform for unsupervised surface defect detection. In: *2010 IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*, pp. 295–302. IEEE (2010)
37. Bai, X., Fang, Y., Lin, W., Wang, L.: Saliency-based defect detection in industrial images by using phase spectrum. *IEEE Trans. Ind. Inform.* **10**(4), 2135–2145 (2014)
38. He, S., Jiao, J., Zhang, X., Han, G., Lau, R.W.H.: Delving into salient object subitizing and detection. In: *IEEE International Conference on Computer Vision*, pp. 1059–1067 (2017)
39. Zhang, P., Wang, D., Lu, H., Wang, H., Xiang, R.: Amulet: Aggregating multi-level convolutional features for salient object detection. In: *IEEE International Conference on Computer Vision*, pp. 202–211 (2017)
40. Zhang, P., Wang, D., Lu, H., Wang, H., Yin, B.: Learning uncertain convolutional features for accurate saliency detection. In: *IEEE International Conference on Computer Vision*, pp. 212–221 (2017)
41. Wang, T., Borji, A., Zhang, L., Zhang, P., Lu, H.: A stagewise refinement model for detecting salient objects in images. In: *IEEE International Conference on Computer Vision*, pp. 4039–4048 (2017)
42. Oh, S.J., Benenson, R., Khoreva, A., Akata, Z., Fritz, M., Schiele, B.: Exploiting saliency for object segmentation from image level labels. In: *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition*, pp. 4410–4419 (2017)
43. Simon, M., Yang, G., Darrell, T., Denzler, J., Rodner, E.: Generalized orderless pooling performs implicit salient matching. In: *IEEE International Conference on Computer Vision*, pp. 4970–4979 (2017)

44. Jonathan, L., Evan, S., Trevor, D.: Fully convolutional networks for semantic segmentation. *IEEE Conference on Computer Vision and Pattern Recognition. CVPR*, pp. 3431–3440. IEEE, Boston, MA, USA (2015)
45. Ronneberger, O., Fischer, P., Brox, T.: U-Net: Convolutional Networks for Biomedical Image Segmentation. Springer International Publishing, Berlin (2015)
46. Zhai, Y., Shah, M.: Visual attention detection in video sequences using spatiotemporal cues. In: *ACM International Conference on Multimedia*, pp. 815–824 (2006)
47. Cheng, M.M., Zhang, G.X., Mitra, N.J., Huang, X., Hu, S.M.: Global contrast based salient region detection. In: *Computer Vision and Pattern Recognition*, pp. 409–416 (2011)
48. Hou, X., Zhang, L.: Saliency detection: a spectral residual approach. In: *IEEE Conference on Computer Vision and Pattern Recognition. CVPR '07*, pp. 1–8 (2007)
49. Rudinac, M., Jonker, P.P.: Saliency detection and object localization in indoor environments. In: *International Conference on Pattern Recognition*, pp. 404–407 (2010)
50. Krahenbuhl, P.: Saliency filters: contrast based filtering for salient region detection. In: *Computer Vision and Pattern Recognition*, pp. 733–740 (2012)
51. Zhang, J., Sclaroff, S., Lin, Z., Shen, X., Price, B., Mech, R.: Minimum barrier salient object detection at 80 fps. In: *IEEE International Conference on Computer Vision*, pp. 1404–1412 (2015)
52. Harris, C.: A combined corner and edge detector. *Proc. Alvey Vis. Conf.* **1988**(3), 147–151 (1988)
53. Schaefer, S., Mcphail, T., Warren, J.: Image deformation using moving least squares. In: *ACM SIGGRAPH*, pp. 533–540 (2006)
54. Zhang, M., Li, X., Xu, M., Li, Q.: Image segmentation and classification for sickle cell disease using deformable u-net (2017). *arXiv preprint [arXiv:1710.08149](https://arxiv.org/abs/1710.08149)*
55. Dai, J., Qi, H., Xiong, Y., Li, Y., Zhang, G., Hu, H., Wei, Y.: Deformable convolutional networks. *CoRR*, abs/1703.06211 **1**(2), 3 (2017)
56. Zhu, W., Liang, S., Wei, Y., Sun, J.: Saliency optimization from robust background detection. In: *IEEE Conference on Computer Vision and Pattern Recognition*, pp. 2814–2821 (2014)
57. Tavakoli, H.R., Rahtu, E.: Fast and efficient saliency detection using sparse sampling and kernel density estimation. In: *Scandinavian Conference on Image Analysis*, pp. 666–675 (2011)
58. Jiang, B., Zhang, L., Lu, H., Yang, C., Yang, M.H.: Saliency detection via absorbing Markov chain. In: *IEEE International Conference on Computer Vision*, pp. 1665–1672 (2013)
59. Wang, J., Jiang, H., Yuan, Z., Cheng, M.M., Hu, X., Zheng, N.: Salient object detection: a discriminative regional feature integration approach. *Int. J. Comput. Vis.* **123**(2), 251–268 (2017)



Congying Qiu received her M.S. from Civil Engineering & Engineering Mechanics Department, Columbia University in 2018. Her recent work capitalizes on multidisciplinary application with emphasis on computer vision. In the future, she will focus on interpretable models based on computer vision.



Kui Yuan received his Ph.D degree in Kyushu University Japan at 1988. He was doctoral tutor director of Chinese Academy of Sciences Institute of automation high technology innovation center.



Yibin Huang is currently pursuing the Ph.D degree of Institute of Automation, Chinese Academy of Sciences. His research interests include image processing, computer vision, optical inspection and deep learning.