

See discussions, stats, and author profiles for this publication at: <https://www.researchgate.net/publication/325882869>

Surface Defect Saliency of Magnetic Tile

Conference Paper · June 2018

CITATIONS

0

READS

128

5 authors, including:



Yibin Huang

Institute of automation

6 PUBLICATIONS 0 CITATIONS

SEE PROFILE



Yue Guo

Chinese Academy of Sciences

9 PUBLICATIONS 6 CITATIONS

SEE PROFILE



Congying Qiu

Columbia University

5 PUBLICATIONS 7 CITATIONS

SEE PROFILE



Kui Yuan

Chinese Academy of Sciences

109 PUBLICATIONS 690 CITATIONS

SEE PROFILE

Surface Defect Saliency of Magnetic Tile

Yibin Huang^{1*}, Congying Qiu², Yue Guo³, Xiaonan Wang³, and Kui Yuan³

Abstract—Vision-based detection on surface defects has long postulated in the magnetic tile automation process. In this work, we introduce a real-time and multi-module neural network model called MCuePush U-Net, specifically designed for the image saliency detection of magnetic tile. We show that the model exceeds the state-of-the-art, in which it both effectively and explicitly maps multiple surface defects from low-contrast images. Our model significantly reduces time cost of machinery from 0.5s per image to 0.07s, and enhances saliency accuracy on surface defect detection.

I. INTRODUCTION

Surface defect detection is a core process of filtering unqualified products, however, the procedure can rarely be finished automatically. It is recorded that almost three-quarters of workers are employed to inspect product quality at the magnetic tile factories in Zhejiang Province, China, the largest magnetic tile production base in the world. To relieve human labor, many image processing techniques have been proposed to attempt such examination tasks. There have been several bottlenecks presented in the automatic damage detection for magnetic tiles, including the complexity of texture, the variety of defect shape, and the randomness of illumination conditions on magnetic tiles. The target defects such as blowhole, crack, break, fray are shown in Figure 1.

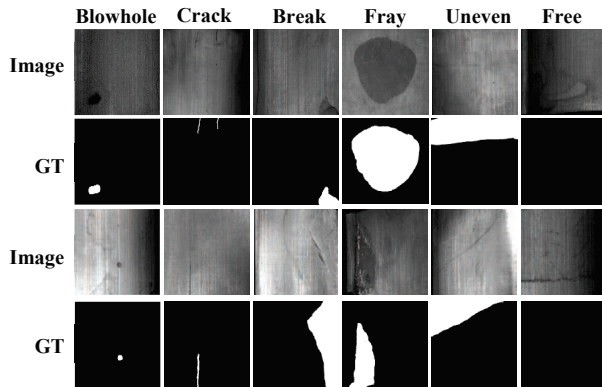


Fig. 1. Examples of magnetic tile surface defects, labeled with pixel-level ground truths (GTs).

This work is supported by National Natural Science Foundation of China No.61421004

*Yibin Huang is from Institute of Automation, Chinese Academy of Sciences, University of Chinese Academy of Sciences, No. 95, Zhongguancun East Road, Beijing, 100190, China huangyibin2014@ia.ac.cn

²Congying Qiu is from Civil Engineering & Engineering Mechanics Department, Columbia University, 500 West 120th Street, New York, NY 10027

³Yue Guo, Xiaonan Wang and Kui Yuan are from Institute of Automation, Chinese Academy of Sciences, No. 95, Zhongguancun East Road, Beijing, 100190, China

In the magnetic tile industry, image-based surface detection mostly concentrates on minimizing the inference of product textures. Embedding denoising techniques (e.g. Wavelet [1], Curvelet [2], and Shearlet [3], [4] transformation) can effectively extract desirable features, but simultaneously lead to long running time. Most notably, the average processing time per image is around 0.5s, which is far from real-time.

Saliency detection models have been explosively and widely used on detecting various surface defects. For example, FT [5], MSS [6] and ITTI [7] -like models are applied in **steel strip surface detection** [8]–[10]; AC [11] -like model are used in weld inspection [12]; PHOT [13] -like model are used in electronic chip inspection [14]. In this sense, it strongly demonstrates the potentials of saliency detection model on surface defect examination of magnetic tiles. At the same time deep neural networks have become dominant on the surface defect detection, achieving state-of-the-art performance on classification tasks [15], [16]. It is also worth noting that the recent progress [17]–[19] in the saliency detection are mostly relies on deep learning. Overall, these explorations indicate that deep learning based saliency detection may be a feasible solution in surface defect inspection.

II. THE MULTI-CUE SALIENCY METHOD

A. Saliency cue

Two main surface defects of magnetic tiles, blowhole and crack, are mainly studied, their characteristics in the grayscale raw images are intensively analyzed. Five important cues useful for Multi-Cue (MCue) saliency construction have been found: (1) reflective intensity of blowhole and crack is weaker than their surrounding environments; (2) corner and the edge responses of blowhole and crack are usually stronger; (3) fray values of defects are relatively significant in the local area; (4) textures in image background are regular; (5) these two defects can easily detected with human visual attention mechanism.

Darker cue. Surface microgeometry of magnetic tile is to changes once defects appear, causing the diffuse reflection of micro-surfaces to change as well. Accordingly, defects such as blowhole and crack look darker than surroundings. However, regular binarization still cannot precisely partition those defects since the illumination has not been normalized in the images. Therefore adaptive binarization is used to calculate darker cues, and it is defines as:

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

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

Strukturtensor cue. The different diffuse reflection between a defect and a non-defect area causes sharp edges or corners around the defect. The corner or edge response of the defects will be stronger than that of the backgrounds. Harris [20] found that the strukturtensor (ST) (structure tensor) can be used for detecting corners and edges. Strukturtensor is a Hessian matrix of the image, and 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 [20] proved that, edge responses will occur when one eigenvalue is large while the other one is small, and corner responses happen if and only if both eigenvalues are large. So let $A = (\lambda_1 - \lambda_2)^2$ represents the edge response an anchor pixe, and $B = |\lambda_1 + \lambda_2|$ represents the corner response of an anchor pixel. Then with equation (3), we get:

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

Finally, the definition of strukturtensor of an image is:

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

where $SalA$ and $SalB$ are computed by equation 4, which represent the edge and corner saliency map of the input image, $\mathcal{N}(\cdot)$ represents normalizing of the saliency map.

B. Cue fusion

In the third cue, we point out that the grey level of blow-holes and cracks are distinguishable from the surroundings, in which the defect color easily grabs attention because of its rarity. The grey levels 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 computational complexity for the network. Based on above consideration, **AC [11]** model is selected to perform this specific task. Specifically, it computes the target pixel quality, as well as measures the Euclidean distance between the value of a selected central pixel and the average values of multiple neighborhoods with different sizes. The color rarities of local regions are thus obtained.

According to the fourth cue we list, the distribution of background textures follows fixed patterns. These textures

are approximately parallel and heavy, producing a lot of noises to the predicted results. PHOT [13] can be used for eliminating these interference of background textures. And BMS [21] is used to detect saliency objects by simulating human visual-attention system. We have designed an accessible, readable, realizable and transferable toolbox, to address the surface defect detection problem in industry. AC, PHOT, BMS and other 11 models have also been assembled in the toolbox, which can be found at <https://github.com/abin24/Saliency-detection-toolbox>.

Based on the above-mentioned considerations, two saliency cues called MCue and MCue2 are proposed. They are constructed as:

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

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

Where S_{BMS} , S_{AC} , S_{ST} , S_{PHOT} , and \mathcal{D} respectively represent saliency maps of BMS, AC, Strukturtensor, and darker cue. W_D, W_{BMS}, W_{PHOT} are weights of \mathcal{D} , BMS, PHOT saliency map, m, n are the normalization constant. Results of MCue fusion results are shown in Figure 2.

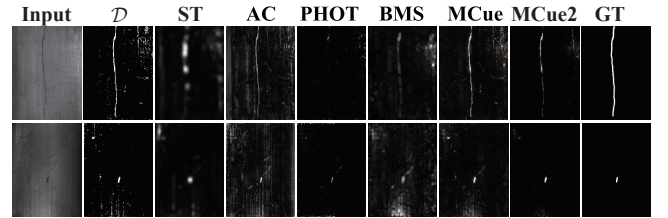


Fig. 2. Saliency map comparison. MCue2 has better fusion than MCue.

Image pixelwise addition interactively complements the saliency regions from each operated image; those regions are thus strengthened. at the same time, image multiplication only strengthens the areas with high saliency value in both images, else areas with low saliency value in either image will be weakened.

We implement pixelwise addition and multiplication on image fusion, in which $W_D = W_{BMS} = W_{PHOT} = 3$, $m = 4$ and $n = 5$. The results are shown in Figure 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 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 W_{PHOT} with a high weight. Simultaneously, direct multiplication likely leads to false detection because of the strong restriction to darker cue \mathcal{D} . We conduct image multiplication with a weight W_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 section IV.

III. U-NET BASED SALIENCY

Encoder-decoder structures generated from Fully Convolutional Network (FCN) [22] are mostly used in cutting-edge saliency detection models [17], [23], [24]. U-Net [25] is a significant extension of FCNs, originally designed for biomedical cell image segmentation. This elegant design not only avoids key information loss during pooling but also enables the model to be learned pixel-to-pixel and end-to-end. Specifically, the convolutional layer in the U-Net encoder part is directly copied and arbitrarily cropped to attain a same size upsampling layer, and skipping the fixed hierarchy connections helps repair details lost from the pooling process. Similar architectures have been applied in many advancing saliency detection models for more accurate saliency maps.

A. Proposed methodology

To identify surface defects of magnetic tiles, a saliency detection model MCuePush U-Net is proposed, and illustrated in Figure 3. Practically, smaller networks run faster and need less computation overhead, so an 98×98 input is used. The input layer is defined as $98 \times 98 \times 3$ including one MCue2 saliency image and two raw images. Compared to a two-channel input, a three-channel input is easier to be used to store the trained images. Meanwhile, Region of Interest (ROI) has been introduced to describe the image content, slides the entire image with input-size stride. 196×196 ROI are cropped from the MCue2 saliency map and raw images, and resized into 98×98 as the network input. The main component of the network architecture is U-Net. In contrast to the U-Net with a full-image input, our architecture effectively reduces the computational complexity and running time.

Softmax is selected as the activation function to present a two-categorical distribution from 0 to 1, and it produces scores corresponding to the foreground and background. Thus the final output has two independent channels in terms of two-dimensional scores. Foreground is particularly focused in the MCuePushU, its predicted probability is:

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

Where $Pred$ is the score or the predicted probability of foreground, f and b respectively represents the foreground and background. To separate the foreground and background from above-mentioned scores, A cross-entropy loss is used.

The word "Push" in MCuePushU refers to an independent structure outside the U-Net, where we herein call it a push network. In the push network, two layers are fully connected, followed by an output layer predicting every surface defect with a bounding box, as shown in Figure 3.

Euclidean algorithm is applied as loss function. Given the labeled center coordinates (x_L, y_L) , width w_L and height

h_L of each defect, the loss function is formulated as:

$$\begin{aligned} \mathcal{L}_{Push} &= ||e_x, e_y, e_w, e_h|| \\ &= (x_L - x_{Pred})^2 + (y_L - y_{Pred})^2 \\ &\quad + (w_L - w_{Pred})^2 + (h_L - h_{Pred})^2 \end{aligned} \quad (9)$$

The label (x_L, y_L, w_L, h_L) is a bounding box, generated from the ground truth of a raw image. $x_{Pred}, y_{Pred}, w_{Pred}, h_{Pred}$ respectively represent the center coordinates, width and height of predicted bounding box.

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 locations and sizes of surface defects. In this designed architecture, most high-level information are gradually embedded using an encoder, key information loss is dedicatedly repaired and prediction accuracy is improved with a decoder. Push network is crucial for helping the U-Net to output desirable saliency maps.

B. Network Training

Labeling pixel-level ground truth, most notably for gray-scale images of magnetic tiles requires extremely heavy workloads, so it becomes implausible to obtain a great amount of labeled data, causing the training process more challenging. Due to this limitation, we consider using strong data augmentation to train our saliency network.

There have existed enormous techniques for image augmentation, and the most commonly seen methods include image rotation, flipping, cropping, and transposition. Since data augmentation on a single small dataset may cause overfitting, to prevent the model from overfitting, an distinct image deformation method using Moving Least Square [26] is introduced. Specifically raw images, together with the corresponding ground truths are used for augmenting. Each image is augmented into 12 using Moving Least Square. Afterwards, these 12 deformed images are augmented into 30 times data size of themselves with regular data augmentations. Therefore $\sim 31,000$ images along with ground truth labels are obtained for network training.

Stochastic Gradient Descent (SGD) with momentum is used as the optimization algorithm in U-Net, where we set momentum term as 0.9, learning rate as 0.0001, and mini-batch size as 8. Multiple losses are involved in this network, when training MCuePushU, the loss functions of U-Net and Push network are simultaneously minimized. One single epoch consists of 400 iterations. However, during the optimizing process, the loss function of Push network converges slower 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 Push network continues being updated in the following 150 iterations.

Essentially, saliency defect detection is a binary classification problem, in which images are classified into two classes: foreground and background. It is worthwhile to mention that most cases the saliency maps are very tiny; therefore foreground pixels class is far fewer than of the background

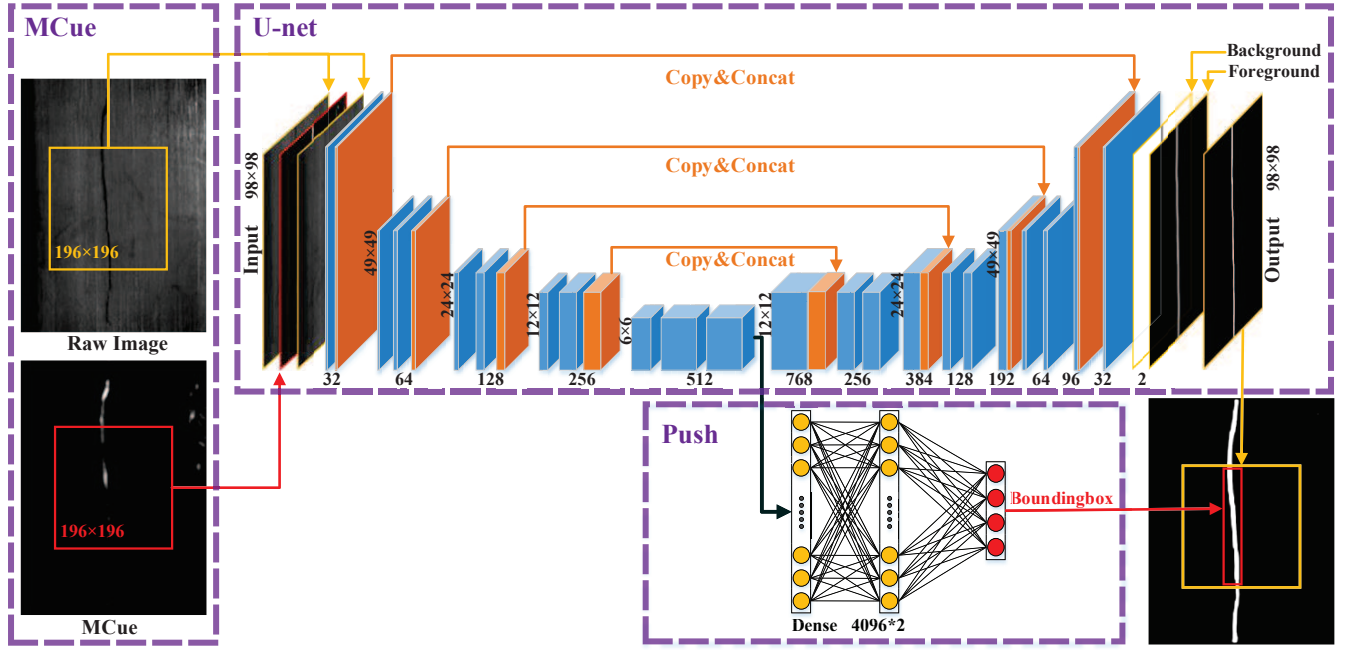


Fig. 3. An architecture overview of MCuePushU with three main components: MCue, U-Net and Push Network.

ones. When the foreground pixels are mistakenly categorized, losses may not be updated any longer because the decrements are too small. Thus, the optimization may be stuck in local minimum, and output maps ultimately turned into all black. In such cases, training process should be restarted to re-initialize parameters. The runtime for 42,000 iterations is ~ 10 hours on the computer equipped with Intel i7 7th GEN CPU, 8G RAM, and one GTX 1050ti Graphics card.

IV. EXPERIMENTS

A. Datasets and evaluation metrics

Datasets. A total of 1344 images are sampled and ROIs of the concerning surface of magnetic tiles are cropped. These image files and their pixel level labels are separated into six datasets according to different defect types: *Blowhole*, *Crack*, *Fray*, *Break*, *Uneven* (grinding uneven), and *Free* (no defects). ROIs of the different magnetic tiles is different, causing varied image sizes. As shown in Figure 1. *Blowhole* and *Crack* impact the qualities of magnetic tiles the most, also have many similar characteristics, so their are merged into a dataset called *Blowhole&Crack*. All the six datasets also merge together as *All* later to test the performance of our proposed model.

Evaluation Metrics. The general approach to quantitatively compare saliency models is comparing the saliency map with the ground-truth. It requires the thresholding saliency maps into binary masks. Saliency maps are binarized at every integer thresholds in the range of $[0, 255]$. Compared to the binary ground-truth mask, precision and recall and true positive rate and false positive rate are recorded. Then the Precision-Recall (PR) curves and Receiver Operating Characteristics (ROC) curves are plot for evaluating saliency detection. Same as [27] Mean Absolute Error (MAE), Area

Under ROC (AUC), and F_β -Measure are also adopted as evaluation metrics. $\beta^2 = 0.3$ in our case in order to attach more importance to the precision. We record the maximum F_β^{Max} from the PR curve, and compute the F_β^{Adp} with the mask from adaptive binarization method same as [27].

B. Performance of MCue model

GMR [28], MBP [29], MC [30], RBD [31] and DRFI [32] are top models of non-deeplearning models in natural scene datasets [27], so they are used to compare to our MCue model.

With the MCue2 model, a output saliency map looks closer to the ground-truth(GT) than MCue in Figure 2. In Figure 4, both the PR and the curve and the ROC curve of MCue2 (the yellow solid curve) is higher than the MCue(the pink solid curve) on the *Blowhole&Crack* dataset. And in Table I, most scores in the row of MCue2 are better than those of MCue. Though MCue and MCue2 is slightly different in fusion, MCue2 seems obviously better than MCue. There for MCue2's Cue fusion method is more reasonable than that of MCue.

It is also conspicuous that, on the *Blowhole&Crack* dataset, MCue2 perform better than all the other models in PR and ROC curve except for models based on U-Net. In Table I, without considering the deep learning based models, the best scores of AUC, MAE and F_β^{Max} are obtained using MCue2, and the best F_β^{Adp} score is achieved with MCue.

C. Improvement in MCuePushU

MCuePush U-Net, namely MCuePushU is used to compare with the original U-Net and the deformable U-Net.

Deformable U-Net [33] is built upon U-Net, deformable convolutional kernels [34] are used to adaptively learn spatial

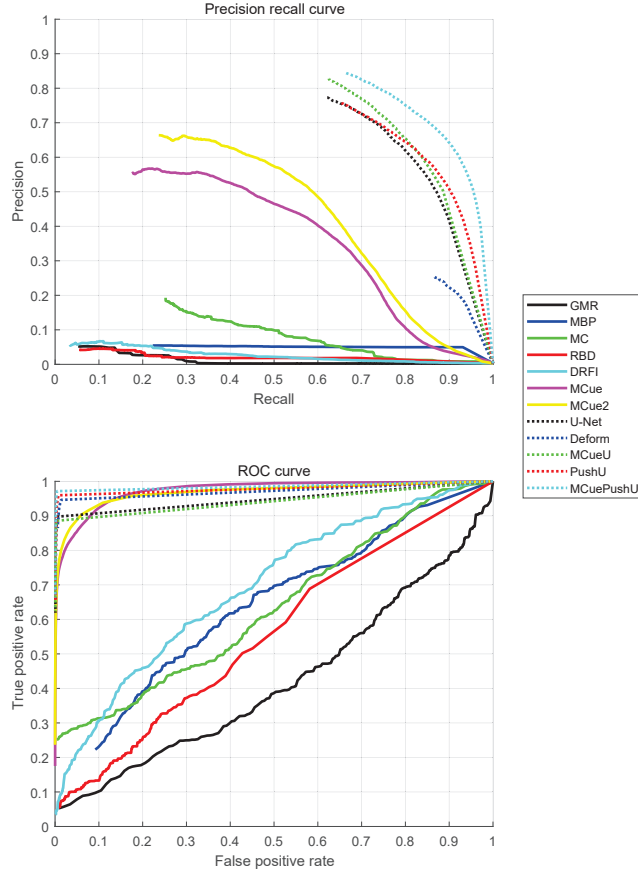


Fig. 4. Model comparison of PR curve (top) and the ROC curve (down) of the *Blowhole&Crack* dataset.

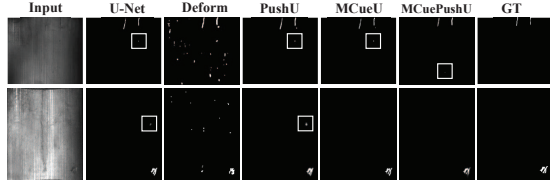


Fig. 5. The improvement of the MCuePushU. White boundingboxes in the images are false alarms, MCuePushU has fewer false alarm.

sampling locations in different sizes. Data augmentation is no longer needed in this model because it learns spatial invariance features during the training [33]. Deformable U-Net achieved better results than standard U-Net in biological image segmentation tasks.

To testify the availability of our proposed components based on MCue and U-Net, experiments respectively on Push, MCue U-Net (MCueU) and Push U-Net (PushU) are conducted for comparisons. Specifically the hyperparameters are set as the same in the experiments comparison, including input size, kernel amount, batch size, learning rate, and iteration number. Data augmentations are still applied to increase the dataset variances.

PR and ROC curves in Figure 4 evidently shows that MCuePushU is dominant over all the other models; the

TABLE I
MODELS COMPARISONS. THE BEST SCORE IN RED, SECOND IN GREEN
AND THIRD IN BLUE.

Model	AUC	MAE	F_{β}^{Max}	F_{β}^{Adp}
GMR	0.420	0.675	0.058	0.095
MBP	0.625	0.435	0.067	0.202
MC	0.633	0.361	0.204	0.267
RBD	0.558	0.130	0.053	0.142
DRFI	0.695	0.340	0.074	0.274
MCue	0.974	0.032	0.490	0.331
MCue2	0.968	0.007	0.560	0.268
U-Net	0.948	0.002	0.732	0.531
Deform	0.967	0.009	0.306	0.495
MCueU	0.942	0.002	0.770	0.535
PushU	0.978	0.003	0.731	0.527
MCuePushU	0.985	0.002	0.795	0.549

performances of PushU and MCueU are both slightly better than U-Net. And in Table I, scores of MCuePushU are the highest. However, result of the deformable U-Net gets is very poor because of overfitting, a large amount of background pixels are wrongly classified into the foreground, as shown in Figure 5 In addition, the processing time of the deformable U-Net is around 0.7s per image, 10 times longer than MCuePushU. McuePushU is also much faster than the average speed (0.5s) of Wavelet [1], Curvelet [2] and Shearlet [3], [4].

MCue is particularly designed based on the *Blow-hole&Crack* dataset, so MCuePushU on other datasets is not tested because MCue has been integrated in MCuePushU. Some failures of PushU on *All* dataset are shown in Figure 6. Specifically, the uneven defect on the left cannot be completely extracted, and on the image with no defect on the right, some polluted areas look very similar to crack or fray, so they are mistakenly classified as defects, and black bounding boxes predicted by the network also suggest the attention of network are focused on the polluted areas.

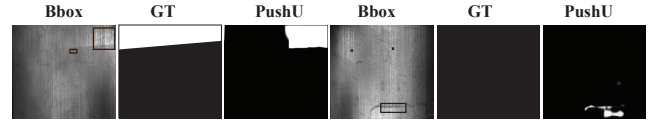


Fig. 6. False detection examples of PushU on *All* dataset, especially for *Uneven*, and dark surface dirt.

V. CONCLUSION

We explore the potential of image saliency in the industrial application: magnetic tile defect detection. The *Blowhole* and *Crack* particularly draws our attention due to its crucial impacts on the magnetic tile manufacturing. We propose a customized model named MCuePushU; this model summarizes a set of dominant cues, then fuses them into 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 performances

as well as meets the demand of real-time inspection process, exceedingly outperforms all other models we test in this article.

There still exist some defects that are hard to be detected. In future, we will focus on improving algorithms to further satisfy comprehensive defect detection.

REFERENCES

- [1] C. Yang, P. Liu, G. Yin, H. Jiang, and X. Li, "Defect detection in magnetic tile images based on stationary wavelet transform," *NDT & E International*, vol. 83, pp. 78–87, 2016.
- [2] X. Li, H. Jiang, and G. Yin, "Detection of surface crack defects on ferrite magnetic tile," *NDT & E International*, vol. 62, pp. 6–13, 2014.
- [3] L. Xie, L. Lin, M. Yin, L. Meng, and G. Yin, "A novel surface defect inspection algorithm for magnetic tile," *Applied Surface Science*, vol. 375, pp. 118–126, 2016.
- [4] C. Yang, P. Liu, G. Yin, and L. Wang, "Crack detection in magnetic tile images using nonsubsampling shearlet transform and envelope gray level gradient," *Optics & Laser Technology*, vol. 90, pp. 7–17, 2017.
- [5] R. Achanta, S. Hemami, F. Estrada, and S. Susstrunk, "Frequency-tuned salient region detection," in *Computer Vision and Pattern Recognition, 2009. CVPR 2009. IEEE Conference on*, 2009, pp. 1597–1604.
- [6] R. Achanta and S. Susstrunk, "Saliency detection using maximum symmetric surround," in *IEEE International Conference on Image Processing*, 2010, pp. 2653–2656.
- [7] L. Itti, C. Koch, and E. Niebur, "A model of saliency-based visual attention for rapid scene analysis," *IEEE Trans.*, vol. 20, no. 11, pp. 1254–1259, 1998.
- [8] K. C. Song, S. P. Hu, Y. H. Yan, and J. Li, "Surface defect detection method using saliency linear scanning morphology for silicon steel strip under oil pollution interference," *Isij International*, vol. 54, no. 11, pp. 2598–2607, 2014.
- [9] K. Song and Y. Yan, "Micro surface defect detection method for silicon steel strip based on saliency convex active contour model," *Mathematical Problems in Engineering*, 2013, (2013-12-22), vol. 2013, no. 8, pp. 1–13, 2013.
- [10] S. Guan, "Strip steel defect detection based on saliency map construction using gaussian pyramid decomposition," *Transactions of the Iron & Steel Institute of Japan*, vol. 55, no. 9, pp. 1950–1955, 2015.
- [11] R. Achanta, F. Estrada, P. Wils, and S. Susstrunk, "Salient region detection and segmentation," in *International Conference on Computer Vision Systems*, 2008, pp. 66–75.
- [12] M. B. Gharsallah and E. B. Braïek, "Weld inspection based on radiography image segmentation with level set active contour guided off-center saliency map," *Advances in Materials Science & Engineering*, vol. 2015, no. 11, pp. 1–10, 2015.
- [13] D. Aiger and H. Talbot, "The phase only transform for unsupervised surface defect detection," in *Computer Vision and Pattern Recognition (CVPR), 2010 IEEE Conference on*. IEEE, 2010, pp. 295–302.
- [14] X. Bai, Y. Fang, W. Lin, and L. Wang, "Saliency-based defect detection in industrial images by using phase spectrum," *Industrial Informatics IEEE Transactions on*, vol. 10, no. 4, pp. 2135–2145, 2014.
- [15] Y.-J. Cha, W. Choi, and O. Büyükoztürk, "Deep learning-based crack damage detection using convolutional neural networks," *Computer-Aided Civil and Infrastructure Engineering*, vol. 32, no. 5, pp. 361–378, 2017.
- [16] J. Masci, U. Meier, D. Ciresan, J. Schmidhuber, and G. Fricout, "Steel defect classification with max-pooling convolutional neural networks," in *Neural Networks (IJCNN), The 2012 International Joint Conference on*. IEEE, 2012, pp. 1–6.
- [17] S. He, J. Jiao, X. Zhang, G. Han, and R. W. H. Lau, "Delving into salient object subitizing and detection," in *IEEE International Conference on Computer Vision*, 2017, pp. 1059–1067.
- [18] V. Ramanishka, A. Das, J. Zhang, and K. Saenko, "Top-down visual saliency guided by captions," in *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*, vol. 1, no. 2, 2017, p. 7.
- [19] S. Joon Oh, R. Benenson, A. Khoreva, Z. Akata, M. Fritz, and B. Schiele, "Exploiting saliency for object segmentation from image level labels," in *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition*, 2017, pp. 4410–4419.
- [20] C. Harris, "A combined corner and edge detector," *Proc Alvey Vision Conf*, vol. 1988, no. 3, pp. 147–151, 1988.
- [21] J. Zhang and S. Sclaroff, "Saliency detection: A boolean map approach," in *IEEE International Conference on Computer Vision*, 2013, pp. 153–160.
- [22] X. Wu, "Fully convolutional networks for semantic segmentation," *Computer Science*, 2015.
- [23] P. Zhang, D. Wang, H. Lu, H. Wang, and R. Xiang, "Amulet: Aggregating multi-level convolutional features for salient object detection," in *IEEE International Conference on Computer Vision*, 2017, pp. 202–211.
- [24] P. Zhang, D. Wang, H. Lu, H. Wang, and B. Yin, "Learning uncertain convolutional features for accurate saliency detection," in *IEEE International Conference on Computer Vision*, 2017, pp. 212–221.
- [25] O. Ronneberger, P. Fischer, and T. Brox, *U-Net: Convolutional Networks for Biomedical Image Segmentation*. Springer International Publishing, 2015.
- [26] S. Schaefer, T. Mcphail, and J. Warren, "Image deformation using moving least squares," in *ACM SIGGRAPH*, 2006, pp. 533–540.
- [27] A. Borji, M. M. Cheng, H. Jiang, and J. Li, "Salient object detection: A benchmark," *IEEE Transactions on Image Processing*, vol. 24, no. 12, pp. 5706–5722, 2015.
- [28] C. Yang, L. Zhang, H. Lu, R. Xiang, and M. H. Yang, "Saliency detection via graph-based manifold ranking," in *IEEE Conference on Computer Vision and Pattern Recognition*, 2013, pp. 3166–3173.
- [29] J. Zhang, S. Sclaroff, Z. Lin, X. Shen, B. Price, and R. Mech, "Minimum barrier salient object detection at 80 fps," in *IEEE International Conference on Computer Vision*, 2015, pp. 1404–1412.
- [30] B. Jiang, L. Zhang, H. Lu, C. Yang, and M. H. Yang, "Saliency detection via absorbing markov chain," in *IEEE International Conference on Computer Vision*, 2013, pp. 1665–1672.
- [31] W. Zhu, S. Liang, Y. Wei, and J. Sun, "Saliency optimization from robust background detection," in *IEEE Conference on Computer Vision and Pattern Recognition*, 2014, pp. 2814–2821.
- [32] J. Wang, H. Jiang, Z. Yuan, M. M. Cheng, X. Hu, and N. Zheng, "Salient object detection: A discriminative regional feature integration approach," *International Journal of Computer Vision*, vol. 123, no. 2, pp. 251–268, 2017.
- [33] M. Zhang, X. Li, M. Xu, and Q. Li, "Image segmentation and classification for sickle cell disease using deformable u-net," *arXiv preprint arXiv:1710.08149*, 2017.
- [34] J. Dai, H. Qi, Y. Xiong, Y. Li, G. Zhang, H. Hu, and Y. Wei, "Deformable convolutional networks," *CoRR, abs/1703.06211*, vol. 1, no. 2, p. 3, 2017.