A Generic Deep-Learning-Based Approach for Automated Surface Inspection

Ruoxu Ren, Terence Hung, and Kay Chen Tan, *Fellow, IEEE*

***Abstract*—Automated surface inspection (ASI) is a challenging task in industry, as collecting training dataset is usually costly and related methods are highly dataset-dependent. In this paper, a generic approach that requires small training data for ASI is proposed. First, this approach builds classifier on the features of image patches, where the features are transferred from a pretrained deep learning network. Next, pixel-wise prediction is obtained by convolving the trained classifier over input image. An experiment on three public and one industrial data set is carried out. The experiment involves two tasks: 1) image classification and 2) defect segmentation. The results of proposed algorithm are compared against several best benchmarks in literature. In the classification tasks, the proposed method improves accuracy by 0.66%–25.50%. In the segmentation tasks, the proposed method reduces error escape rates by 6.00%–19.00% in three defect types and improves accuracies by 2.29%–9.86% in all seven defect types. In addition, the proposed method achieves 0.0% error escape rate in the segmentation task of industrial data.**

***Index Terms*—Automated surface inspection (ASI), deep learning (DL), feature transferring, segmentation.**

I. Introduction

Surface inspection using computer vision techniques is an important task in industry. Traditionally, surface inspection is performed manually. In most cases, it is highly subjective and time-consuming. To overcome the limitations of human inspection, automated surface inspection (ASI) techniques are to assist or replace human decisions.

In the industry, surface defects are local anomalies in homogeneous textures. There is a wide range of surface textures in ASI, such as stone [1], textile [2], wood [3], metal [4], and ceramic tiles [5]. Depending on the properties of surface texture, ASI methods can be divided into four categories [6]: 1) structural; 2) statistical; 3) filter-based; and 4) model-based approaches. Structural methods model the texture primitives and displacements. This approach is usually applied to repetitive patterns such as textile [7], fabrics [8], and leather [9]. Popular structural approaches include primitive measurement [10], edge features [9], skeleton representation [7], and morphological operations [8], [10]. The second approach is statistical methods which measure the distribution of pixel values. The statistical approach is efficient for stochastic textures, such as ceramic tiles, castings, and wood. Popular statistical methods include histogram-based method [11], local binary pattern (LBP) [12], and co-occurrence matrix [13]. The third approach is filter-based methods that apply filter banks on texture images. The filter-based methods can be divided into spatial-domain [14], frequency-domain [15], and spatial-frequency domain [16], [17]. Finally, model-based approaches construct representations of images by modeling multiple properties of defects [5], [18], [19].

Despite the large varieties of ASI methods, these techniques all aim at constructing templates or features for images. As a result, the performance of ASI technique depends on how well the representations can model the defect properties. For ASI-related tasks, there is no explicit guideline for choosing optimal representations. As such, human expertise is the key to the success of ASI. It is challenging to provide generic ASI methods that apply to wide range of textures.

In recent years, deep learning (DL) methods have been achieving good performance on image related tasks such as object recognition [20]. However, it is rarely applied in the field of ASI due to two reasons. First, surface defect datasets are normally too small to train DL networks. Large amount of data is required for DL methods to avoid overfitting. Collecting defect images at such scale is costly because of the exceptionally low probability of defects occurrence, and intensive manual work is required to label all the images. The second challenge is that ASI requires precise identification of defect area, while conventional DL networks are usually for image classification or bounding box generation [21], [22].

To deal with the two challenges, a generic approach for ASI is proposed in this paper. The proposed method extracts patch feature using pretrained DL network, generates the defect heat map based on patch features, and predicts the defect area by thresholding and segmenting the heat map. The proposed method does not require large-scale training data as the patch features are high-level representations extracted from pretrained model without any training. In addition, the proposed

Manuscript received October 7, 2016; revised January 21, 2017; accepted February 3, 2017. Date of publication February 24, 2017; date of current version February 14, 2018. This work was supported by the Economic Development Board Industrial Postgraduate Fund under Grant R-263-000-A70-592. This paper was recommended by Associate Editor Y. Zhang.

R. Ren is with Applied Technology Group, Rolls-Royce Singapore, Singapore 797575, and the Department of Electrical and Computer Engineering, National University of Singapore, Singapore (e-mail: ren.ruoxu@u.nus.edu).

T. Hung is with Applied Technology Group, Rolls-Royce Singapore, Singapore 797575 (e-mail: terence.hung@rolls-royce.com).

K. C. Tan is with the Department of Computer Science, City University of Hong Kong, Hong Kong (e-mail: kaytan@cityu.edu.hk).

This paper has supplementary downloadable multimedia material available at http://ieeexplore.ieee.org provided by the authors. This material is 1.65 MB in size.

Color versions of one or more of the figures in this paper are available online at http://ieeexplore.ieee.org.

Digital Object Identifier 10.1109/TCYB.2017.2668395

method enables defect area identification by generating defect heat map based on a trained patch-classifier. The experimental results show that the proposed method works well on small dataset, which largely reduces the cost of collecting data, and the proposed approach applies across several types of surface defect datasets.

The organization of this paper is as follows. Section II discusses the related work of this paper. Next Section III describes the methodology for this paper, where the transferred feature is discussed and the proposed ASI method is described in details. Section IV describes the four datasets that are used for evaluation. In Section V, the experimental results are presented. Next, Section VI shows the implementation details. In Section VII, the results are further discussed. Finally, Section VIII concludes this paper.

II. Related Work

The related work of this paper is descried in this section, including the convolutional neural network (CNN) named Decaf and feature transferring from pretrained CNN.

A. Deep Convolutional Neural Network: Decaf

CNN is a type of feed-forward neural network widely applied for image-related analysis. CNNs normally consist of three major layers: 1) convolutional layer; 2) pooling layer; and 3) fully connected layer. The convolutional layer applies a number of filters on the local regions of the input, thus obtaining the feature maps of the input image. Suppose the number of filters is k, W^i denote the weight of filter i, b^i is the bias of filter i, x_s stand for a small patch, σ is the activation function, and the size of input image is $M \times N$. The convolution of x_s given filter i is shown in

$$f_{i,s} = \sigma\left(W^i x_s + b^i\right). \tag{1}$$

By sliding the patch window through the image with patch size $a \times b$ and stride size s, the convolutional output with size $k \times \lceil[(M-a+1)/s]\rceil \times \lceil[(N-b+1)/s]\rceil$ can be obtained, where $\lceil\cdot\rceil$ is the ceiling function.

The pooling layer performs down-sampling operation along spatial dimensions of the input. For example, max pooling of image patch x_s is simply

$$\text{pool}_s = \max(x_s). \tag{2}$$

The pooling layer is normally applied after the convolutional layer to reduce the feature dimension and to avoid over-fitting problem. For instance, max pooling with input size $M \times N$ and patch size $c \times d$ produces output with size $\lceil[(M-1)/c]\rceil \times \lceil[(N-1)/d]\rceil$.

The fully connected layers normally constitute the last few layers of a CNN, computing the class scores. A deep CNN normally consists of alternating convolutional and pooling layers, followed by fully connected layers. The CNN structure is widely adopted in computer vision tasks, such as object recognition, scene classification, and video analysis [20]–[23].

Among the various CNNs, Decaf [23] is one of the most popular model for object recognition. It is an open-source deep CNN with five convolutional, three pooling and three fully connected layers. The structure of Decaf is shown in Fig. 1. Decaf is trained to predict 1000 categories of objects in the ImageNet challenge. The trained weights and model structure are shared on the authors' website.[1]

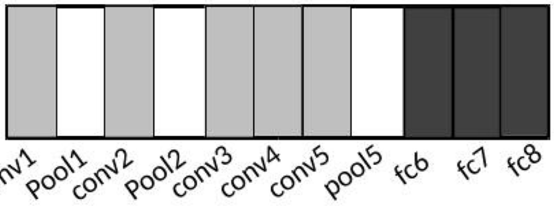


Fig. 1. Decaf layer structure. Conv$_i$: convolutional layers, pool$_i$: pooling layers, and fc_i: fully connected layers.

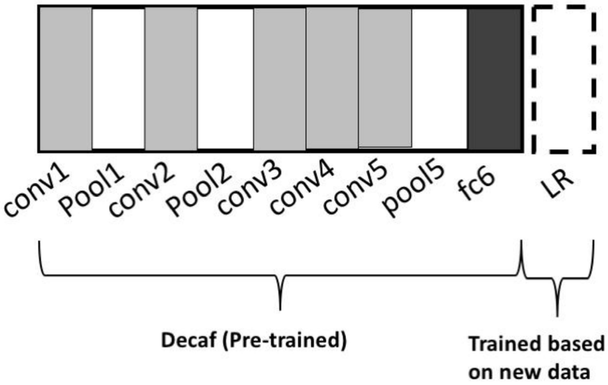


Fig. 2. Example of feature transferring from Decaf and classifying new data based on logistic regression.

B. Feature Transferring From Deep CNN

Although Deep CNNs like Decaf is proven highly efficient for image-related applications, not many people train deep CNNS from scratch in practice due to the large amount of data required for training. Instead, many researchers first train a deep CNN on large-scale image data, and the trained weights are applied as a feature extractor to the small data in another domain [24], [25]. It is normally performed by removing the last few layers of the trained CNN.

Among the wide range of pretrained deep CNNs, the weights of Decaf are often reused as feature extractor. Donahue *et al.* [23] shows that the intermediate layers of the trained Decaf model serve as generic image features. Two intermediate layers are tested as image feature, fc6, and fc7. With the pretrained weights, the transferred feature is obtained by feeding the image into input layer and retrieving the output from the intermediate layer (fc6 and fc7 of Decaf). Fig. 2 shows an example of applying feature transferring for image classification. Donahue *et al.* [23] shows the transferred features are effective for object recognition, scene recognition, subcategory recognition, and domain adaptation. In addition, [26] shows such feature transferring is also efficient for texture images. Furthermore, both [23] and [26] show the

[1] http://daggerfs.com/

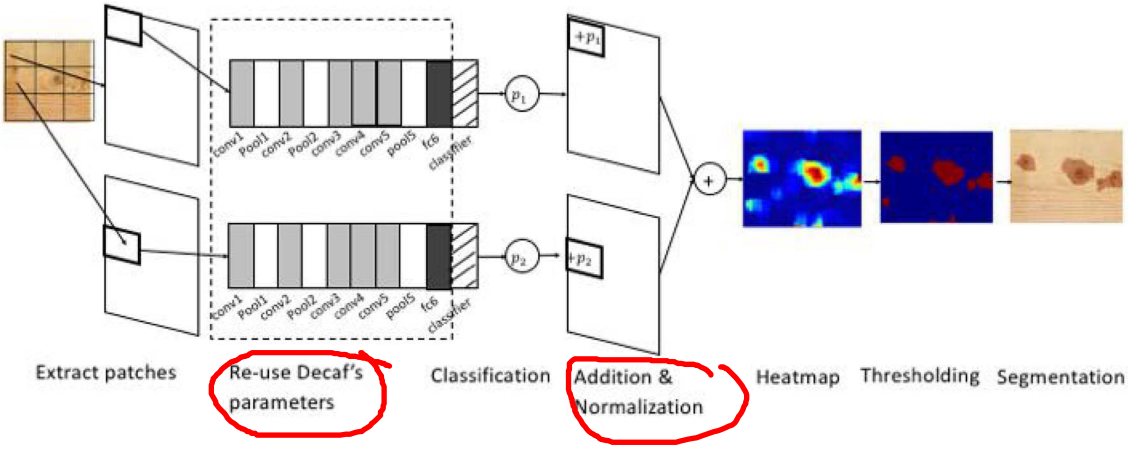


Fig. 3. Structure of proposed method.

transferred features are high-level representations, as the transferred features from different classes are well-separated using clustering methods and they outperform several state-of-art computer vision methods in diverse tasks.

In terms of the quality of fc6 and fc7 as image representation, fc6 is a more generalized image feature than fc7 and the former outperforms the latter in majority of image-related tasks [23].

III. Proposed Generic ASI Method

In this section, the details of proposed ASI method are discussed. There are two phases in the proposed method. The first phase includes supervised training of patch classifier. Based on the trained classifier, the second phase is to segment defect area given an input image. The overall flowchart of the proposed method is shown in Fig. 3.

A. Training of Patch Classifier

In the classifier training stage, a number of image patches are extracted from the training data. The patch classifier is trained based on transferred features from Decaf. The details of patch classifier training are as follows.

1) Patch Extraction: The training patches are collected similarly as the CNN. Suppose patch size is $m \times m$ and the stride size is s. Number of patches extracted from an image with size $M \times N$ is $\lceil[(M - m + 1)/s]\rceil * \lceil[(N - m + 1)/s]\rceil$. The training labels of patches correspond to the defects that appear within the patch. The patches are labeled normal if there are no defect within the patches.

2) Feature Extraction: After collecting the training patches, patch features are extracted using feature transferring as what is described in Section II-B. The intermediate layer fc6 is selected as the feature extractor, as it is proven efficient for wide ranges of computer vision tasks [23], [26].

3) Classifier Selection: Based on the transferred features of training patches, a classifier is trained. The classifier is chosen to be multinomial logistic regression (MLR), as it is naturally close to Softmax regression used in the classification stage of Decaf.

Algorithm 1 HM Generation Algorithm

```
Input ← $Im, m, s$
procedure HM–Generation
    $\{P_1, \cdots, P_{N_{patch}}\} = ConvolvePatches(Im, m, s)$
    $\mathcal{C} \leftarrow 0_{M \times N}$
    $\mathcal{HM}_{1,\cdots,N_d} \leftarrow 0_{M \times N}$
    for each $i \in \{1, \cdots, \lceil \frac{M}{s} \rceil * \lceil \frac{N}{s} \rceil\}$ do
        $\overrightarrow{feat} \leftarrow Decaf_{fc6}(P_i)$
        $\vec{p} \leftarrow Classifier(\overrightarrow{feat})$
        $\mathcal{C}(P_i) \leftarrow \mathcal{C}(P_i) \oplus 1$
        for each $j \in N_d$ do
            $\mathcal{HM}_j(P_i) \leftarrow \mathcal{HM}_j(P_i) \oplus \overrightarrow{p(j)}$
        end for
    end for
    for each $j \in N_d$ do
        $\mathcal{HM}_j \leftarrow \mathcal{HM}_j \oslash \mathcal{C}$
    end for
end procedure
Output ← $\mathcal{HM}_{1,\cdots,j}$
```

B. Segmentation Framework

The second phase is the defect segmentation stage. This stage includes patch extraction, feature extraction, classification, and heatmap generation and prediction.

1) Patch Extraction: The first step in segmentation framework is the patch extraction. The same process is performed and same set of parameters (m, n, and s) are applied as that in Section III-A1.

2) Feature Extraction: The second step is to extract features of the patches obtained. Similarly, all the patches are fed into Decaf and the outputs from fc6 represent the patch features.

3) Classification: In the third stage, the trained classifier is stacked after the nine layers from Decaf (conv1 to fc6). The classifier outputs the probabilities of an input image being in different classes.

4) Heatmap Generation and Prediciton: The next stage is the heatmap generation and defect prediction. Suppose the number of defect types is N_d, the notation for the ith patch is P_i, the coordinates for four corners of P_i is $\{(x_{\{i,1\}}, y_{\{i,1\}}),$

$(x_{\{i,2\}}, y_{\{i,1\}}), (x_{\{i,1\}}, y_{\{i,2\}}), (x_{\{i,2\}}, y_{\{i,2\}})\}$, and the prediction from the classifier is $\vec{p}_i = (\text{prob}_{\{i,1\}}, \text{prob}_{\{i,2\}}, \ldots, \text{prob}_{\{i,N_d\}})$. For each defect class, a heat map is obtained by iteratively adding the probabilities pixel-wise. The pseudo-code for obtaining heat maps is shown in Algorithm 1. In the algorithm, $0_{x\times y}$ denotes zero matrix of size $x \times y$, $\oplus$ denotes element-wise addition, and $\oslash$ denotes element-wise division. A count matrix $\mathcal{C}$ is updated in this algorithm for normalizing the heat maps. As such, the values of normalized heat maps obtained are from 0 to 1.

After obtaining the heat map for each class, Otsu's method [27] is applied for binarization. Otsu's method binarizes an image by obtaining an optimal threshold which minimizes intraclass variance. Suppose the image pixels are represented using L gray-level histogram and p_i is the normalized histogram value of gray level i. The objective of Otsu's method is

$$\min_{T} \ \sigma^2_{\text{intra}}(T) = n_B(T)\sigma^2_B(T) + n_A(T)\sigma^2(T) \tag{3}$$

where

$$n_B(T) = \sum_{i=0}^{T-1} p_i$$

$$n_A(T) = \sum_{i=T}^{L-1} p_i$$

$$\sigma^2_B(T) = \text{variance of pixels below threshold}$$

and

$$\sigma^2_A(T) = \text{variance of pixels above threshold.}$$

After obtaining the binarized image, Felzenswalb's segmentation [28] is applied to refine the defect areas. This algorithm is a graph-based method that groups pixels by their color similarities. The purpose of applying Felzenswalb's segmentation is to remove small disconnected segments from the thresholding step and connect large potential defect areas. Suppose $G = (V, E)$ is an undirected graph, where V is the set of pixels, E is the set of edges, $w(v_i, v_j)$ represents the weight between v_i and v_j, and C_i represents ith region of the segmentation. Let D represent the predicate for determining the boundary of segmentation. D is as shown in

$$D\big(C_i, C_j\big) = \begin{cases} \text{true} & \text{if } \text{Dif}\big(C_i, C_j\big) > M\text{Int}\big(C_i, C_j\big) \\ \text{false} & \text{otherwise} \end{cases} \tag{4}$$

where

$$Dif\big(C_i, C_j\big) = \min_{v_n \in C_i, v_m \in C_j, (v_n, v_m) \in E} w(v_n, v_m)$$

$$\text{Int}(C) = \max_{e \in MST(C,E)} w(e)$$

$$M\text{Int}\big(C_i, C_j\big) = \min\big(\text{Int}(C_i) + \tau(C_i), \text{Int}\big(C_j\big) + \tau\big(C_j\big)\big)$$

$$\tau(C) = k/|C|.$$

There are two parameters to choose in this algorithm. The first one is σ, which is applied to smooth image with Gaussian filter. Following what is described in [28], σ is set to be 0.8. The second parameter is k, which is used to compute thresholds. Large k leads to large component in segmentation result. The selection of parameters are discussed in latter parts of this section.

Algorithm 2 Defect Segmentation

```
Input ← $Im$, $\sigma$, and $k$
procedure SURFACE–INSPECTION
    $\mathcal{HM}_{1,2,\cdots N_d}$ = HM–GENERATION ($Im$)
    for each $j \in (1, 2, \cdots, N_d)$ do
        $\mathcal{B}_j \leftarrow$ Otsu's ($\mathcal{HM}_j$)
        $Regions \leftarrow$ Felzenszwalb's ($\mathcal{B}_j, \sigma, k$) [28]
        for each $R \in Regions$ do
            if $average(HM(R)) < \frac{1}{N_d}$ then
                Remove $R$ from $Regions$
            end if
        end for
    end for
end procedure
Output ← $Regions$
```

Since heat maps are always generated for every type of defect, a threshold is set after the segmentation results to remove the predicted areas with low defect probabilities. The final predictions are thus obtained. The defect regions prediction is shown in Algorithm 2.

C. Parameter Selection

1) Parameters for Heatmap Generation: There are four parameters for heatmap generation, patch size $m \times m$, stride size s. The selection of m depends on the domain knowledge for the size of defect. For example, if the defect size is around 50×50, choosing patch size 100×100 leads to imprecise heat maps, and patch size 5×5 might not include enough information for the defect. As a result, the patch size is selected roughly the same as the size of defect. In terms of parameter s, high value leads to imprecise heatmaps and low value results in high computational cost. In the proposed work, $s = (1/5)m$ is applied across all experiments.

2) Parameters for Defect Area Prediction: To predict the defect area, both Otsu's thresholding method and Felzenzwalb's segmentation are involved. The threshold determination of Otsu's method is fully automatic based on the intensities of input image. However, there are two parameters in Felzenswalb's segmentation, σ and k. σ is the parameter for Gaussian filtering of the image preprocessing stage. σ is always chosen as 0.8, as it does not lead to significant change to the image while removing noises according to [28]. Another parameter k is related to the scale of observation. A larger k results in larger components in the segmentation. This parameter is chosen according to the defect area. According to [28], $k = 200$ for defect size around 20×20, $k = 500$ for defect size around 50×50, and so on.

IV. Data Description

To test if the proposed method could be generalized, a wide range of surfaces with distinctive textures is selected in this experiment. The datasets include texture defects on the hot-rolled strip [29], surface flaws on X-Ray images of

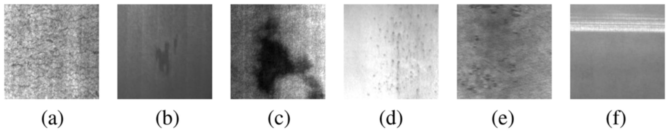


Fig. 4. Examples of NEU Surface Defect Database. (a) Cr. (b) In. (c) Pa. (d) PS. (e) RS. (f) SC.

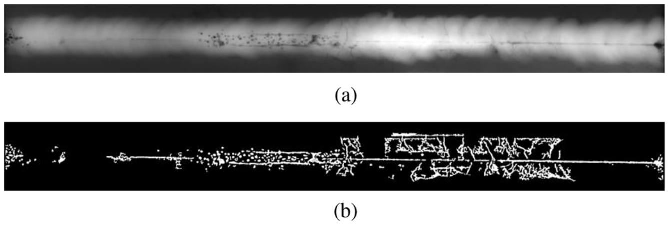


Fig. 5. Example of weld defect database. (a) Original image. (b) Binary ground truth.

metal pipe [30], color and texture defects on lumber [3], and microstructure defects of titanium alloy. These datasets covered texture defects, color defects, and microstructure defects. The details of these datasets are described as follows.

A. NEU Surface Defect Database

The first public database is the NEU surface defect [29]. The database includes six types of surface defects from the hot-rolled steel strip. The defects include crazing (Cr), inclusion (In), pathces (Pa), pitted surface (PS), rolled-in scale (RS), and scratches (SC). There are 300 samples from each class of defect. Examples of defect images are shown in Fig. 4. The labels of all the images are given in the database, while the ground truth for defect areas is not provided.

B. Weld Defect Database

The second public dataset is from GDXray images [30]. This dataset includes X-Ray images of five groups, which are castings, welds, baggages, nature, and settings. The dataset used in this paper is welds, since it is related to surface inspection while the others are not. The welds dataset include radiographs of metal pipes. Ten images with pixel-wise ground truth are provided in the database. Examples of the images are shown in Fig. 5.

C. Wood Defect Database

The last public database in the experimental study is the wood defect database [3] for ASI of lumber. There are two subsets inside the database. One is manually labeled images of defects, which includes different types of wood knots. Four hundred and thirty eight images from seven types of knot defects are provided in this subset. Sample images are shown in Fig. 6. The other subset of this database is wood board images where the ground truth is provided by bounding boxes. Eight hundred and thirty nine board images are provided and 18 types of defects are included. Some examples are shown in Fig. 7.

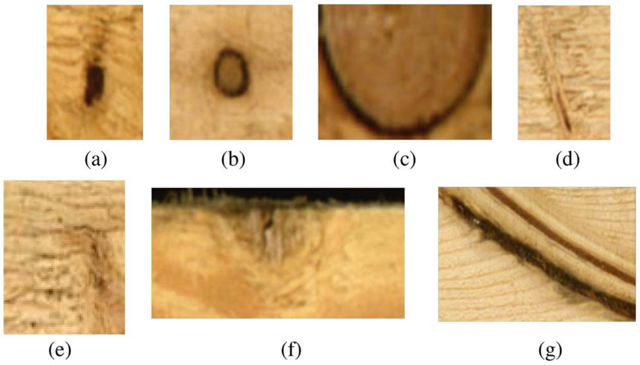


Fig. 6. Sample images in wood defect database (subset 1). (a) Decayed knot. (b) Dry knot. (c) Encased knot. (d) Leaf knot. (e) Sound knot. (f) Edge knot. (g) Horn knot.

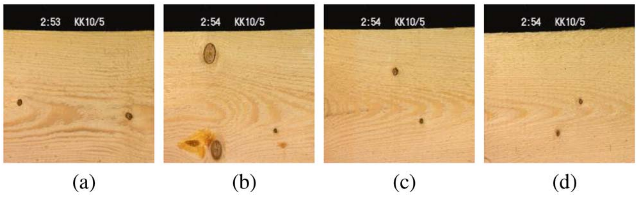


Fig. 7. Sample images of wood defect database (subset 2). (a) Image with dry knot (b) Image with dry knot, resin, resin pocket and sound knot. (c) image with dry knot. (d) Image with dry knot.

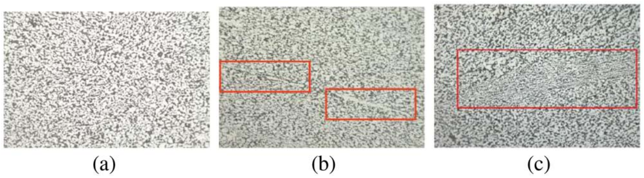


Fig. 8. Examples of micro-structured data. (a) Normal. (b) Defect. (c) Defect.

D. Micro-Structure Defect Dataset

The last dataset is an industrial dataset from Rolls-Royce. The images are collected from titanium fan blades using a microscope and the dimension of images is 1944×2580. The training set includes 49 defect images and ten normal images. The test set includes ten defect images and ten normal images. The defect areas are labeled using bounding boxes. Some examples are shown in Fig. 8.

V. Experimental Results

In this section, experimental results are shown. The evaluation consists of two parts. The first part is to assess the transferred feature, where classifications of defect images are carried out. The second part is to evaluate the proposed segmentation framework.

For the classification results, two general texture feature extraction methods are applied across all public dataset for benchmarking, which are multiresolution local binary patterns (MLBP) [31] and gray level co-occurrence matrix (GLCM) [32]. The parameters chosen for these methods are according to [12] and [31]. For the MLBP, the image feature is the joint distribution of gray-scale rotation invariant local binary pattern and rotation invariant variance measure.

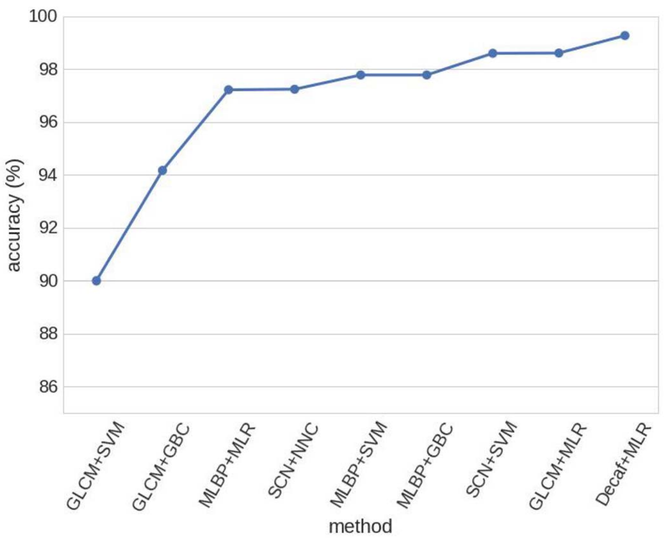


Fig. 9. Classification results of nine methods on NEU surface defect database. Accuracy of proposed method (Decaf+MLR) is 99.27% and that of best-performing benchmark (GLCM+MLR) is 98.61%.

According to [31], features at multiple pairs of neighbors and radius are combined, which are $\{(8, 1), (16, 2), (24, 3)\}$. This method has been proven to perform well on a wide range of texture datasets. Hence, it is selected as a benchmark method. For GLCM, the angles chosen are $\{0°, 45°, 90°, 135°\}$ and the distance set is $\{1, 2, 4, 8, 16\}$. The texture features are a combination of angular second moment, inverse difference moment, entropy, correlation, contrast, and energy [33].

Apart from the general texture features, state-of-art hand-crafted features for individual dataset are also selected as benchmarks. It is to compare the performance of the transferred feature with individually designed algorithms.

For the segmentation evaluation, a number of dataset-dependent ASI methods are compared against the proposed framework.

A. Evaluation of Transferred Features

Features from the pretrained DL network are selected as image representation. The classification performances on the features are evaluated based on the three datasets described in Section IV. The accuracies reported in this section are all based on fivefold cross validation unless otherwise stated in the dataset.

As to individual dataset, some dataset-specific feature extraction and classification methods are tested. The classifier includes gradient boost classifier with loss of log-likelihood, support vector machine (SVM), logistic regression (LR), and MLR. Each test phase consists of 20 runs and the average accuracy is reported.

1) Classification Results of NEU Surface Defect Database: For the NEU surface defect database, whole images instead of patches are selected for evaluation. It is due to the reason that these images are already zoomed-in to patch scale. Traditional texture feature extraction methods such as MLBP and GLCM are used as benchmarks. In [34], scattering convolution network is proven effective on this dataset. It is thus selected as a benchmark method, in which SVM and nearest neighbor classifier are used. The test set is chosen the same way as described in [34], where for each class 150 images are randomly selected. The results are shown in Fig. 9. It can be shown that transferred feature outperforms all the benchmarks with 99.21% accuracy.

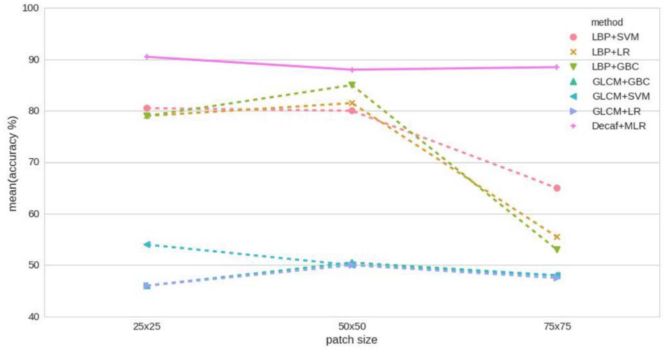


Fig. 10. Classification results of seven methods on weld defect database. The accuracies of proposed method on patch size $25 \times 25, 50 \times 50$, and 75×75 are 90.50%, 88.00%, and 88.50%, respectively. The accuracies of best-performing benchmark on the three patch sizes are 65.00%, 85%, and 80.50%, respectively.

2) Classification of Patches From Weld Defect Database: For the weld images in GDXray dataset, the ground truth is provided for each image. To test on the performance of image feature, patch sizes of $25 \times 25, 50 \times 50$, and 75×75 are evaluated. 200 patches at nondefect area and 200 patches at defect area are selected for each image. The minimum defect size of patches is chosen to be 100 pixel, as the defect area should be of reasonable size to extract the defect information. Five images are selected for generating patches. The patches are evaluated by fivefold cross-validation. The results are shown in Fig. 10.

The results show that transferred feature has the best performance for all three patch sizes. MLBP also performs well in the classification task. However, GLCM is no better than random guess (50%) for this dataset.

3) Classification of Wood Defect Database: The evaluation of transferred feature on wood defect dataset is based on the first subset. The method is compared against [35] and the cross-validation method also follows what was described in [35]. The image feature in [35] is Gabor filters and classifiers include both self-organizing neural network and a feedforward perceptron neural network (FFPNN). The types of defect reported in this methods include encased, leaf, edge, and sound knots.

The results are shown in Fig. 11. Among all the methods, the transferred feature leads to highest accuracy. It outperforms both general texture features (MLBP and GLCM) and hand-crafted classification methods (Gabor filters and FFPNN).

From the above three classification tasks, the transferred feature is a high-level representation of the textures in this experiment. It largely outperforms both the traditional texture features and the hand-crafted features.

B. Evaluation of Defect Segmentation

The classification results show that the transferred feature has high performances over multiple datasets. The segmentation method described in Algorithms 1 and 2 are

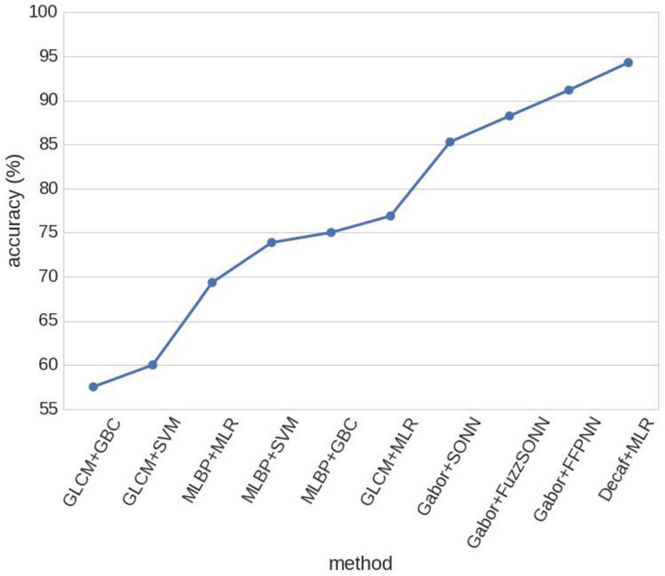


Fig. 11. Classification results of ten methods on wood defect database (subset 1). The accuracy of proposed method is 94.29% and that of best-performing benchmark is 91.17%.

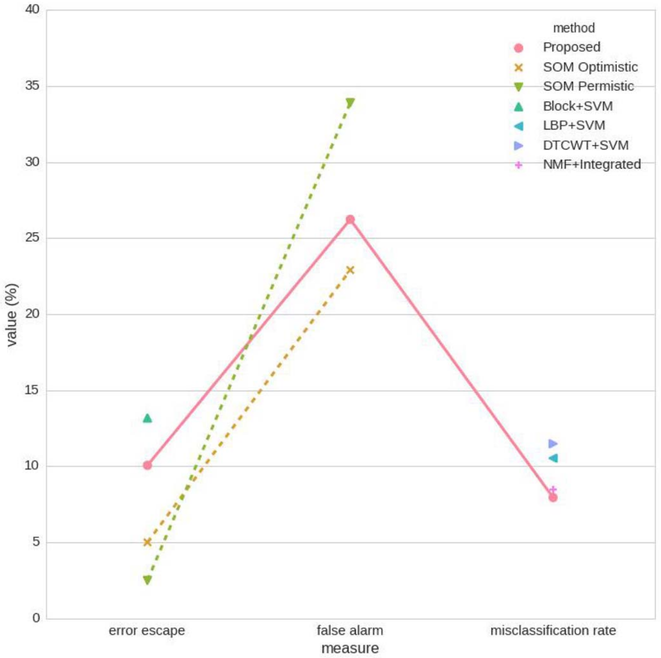


Fig. 12. Segmentation results of wood defect database using proposed method, SOM optimistic [3], SOM permistic [3], block+SVM [37], LBP+SVM [36], DTCWT+SVM [36], and NMF+integrated [36]. The error escape, false alarm, and misclassification rate of proposed method are 10.06%, 26.23%, and 11.6%, respectively. The three measures of best-performing benchmark are 2.50%, 22.90%, and 91.55%, respectively. *Note:* the proposed method is evaluated over the whole dataset, while the benchmarks are evaluated against a subset of defect types.

also implemented on these datasets. For individual dataset, the framework is compared with hand-crafted segmentation methods for the specific dataset. As the ground truth is not available in the NEU surface defect dataset, only the weld defect, wood defect, and microstructure defect datasets are evaluated for segmentation.

1) Segmentation Results on Wood Defect Dataset: For subset 2 of board image in the wood defect database, patch size is selected as 50×50. The parameters for segmentation is chosen as $\sigma = 0.8$ and $k = 500$.

The ground truth provided in this dataset is not pixel-wise. They are bounding boxes defined by four coordinates. As such, the performance evaluation follows what is described in [3] and [36]. The five criterion used are defect detection (D) rate, error-escape (E) rate, false-alarm (F) rate, misclassification rate (M), and accuracy (A). The information required to calculate these metrics are as follows.

1) N_{det}*:* Number of detections generated by the inspection model.
2) N_{lab}*:* Number of true defects.
3) $N_{\text{det\_lab}}$*:* Number of true defects that are detected by the inspection model.
4) N*:* Total number of image patches selected for evaluation (including both defect and normal patches).

From the above definitions, $N_{\text{det\_lab}}$ is the intersection of N_{lab} and N_{det}. Fig. S-4 in the supplementary material visualize the relationship between these parameters.

The defect detection rate is the proportion of real defects that are detected by the inspection model, as follows:

$$D = \frac{N_{\text{det\_lab}}}{N_{\text{lab}}}. \tag{5}$$

The error escape rate is the proportion of defects that are missed by the inspection model as follows:

$$E = 1 - D. \tag{6}$$

The false-alarm rate is represented by the ratio of false positive detection to the total number of detections that are produced by the system as follows:

$$F = \frac{N_{\text{det}} - N_{\text{det\_lab}}}{N_{\text{det}}}. \tag{7}$$

The misclassification rate is the ratio of misclassification to the total number of patches, including both defect and nondefect areas, as follows:

$$M = \frac{\left(N_{\text{lab}} - N_{\text{det\_lab}}\right) + \left(N_{\text{det}} - N_{\text{det\_lab}}\right)}{N}. \tag{8}$$

The accuracy is the percentage of patches that are correctly classified defined as follows:

$$A = 1 - M. \tag{9}$$

In terms of a high-performing ASI method, low error-escape (E), low false-alarm (F), low miss-classification (M) rate, and high accuracy (A) are desired.

The proposed method is compared against the best benchmarks [3], [36], [37]. In [3], the color centiles and LBP is used as feature. Each image is separated into blocks and self-organizing map (SOM) is applied on these image blocks. The threshold between defect and nondefect blocks are manually set. Two methods of choosing threshold is reported in [3], which are optimistic and permistic. The error escape (E) rate and false alarm (F) rate of [3] are reported against ten types of defect. In [37], each image is similarly divided into blocks. The image feature is the percentile color histogram and eigenvector

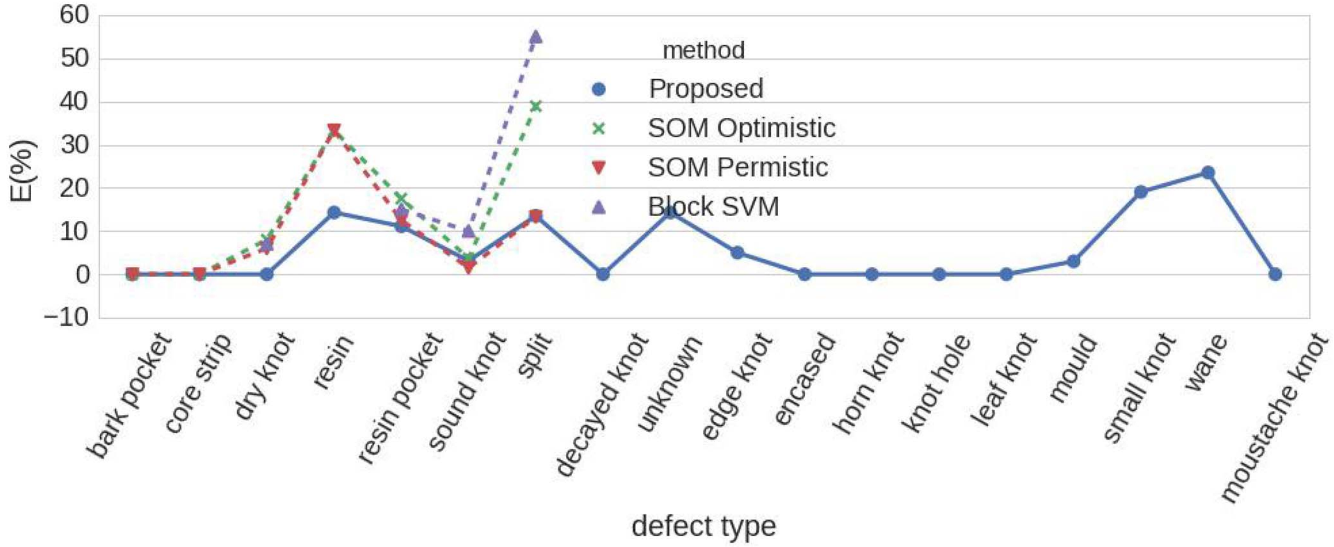


Fig. 13. Error escape rates for wood defect detection using proposed method, SOM optimistic [3], SOM permistic [3], and block SVM [37]. Compared to the benchmark methods, the proposed method reduces error escape of dry knot, resin, and resin pocket by 6.00%, 19.00%, and 1.39%, respectively. In terms of sound knot and split, the error escape rates of proposed method is 1.63% and 0.21% higher than best-performing benchmark. For bark pocket and core strip, the error escape rates of proposed method and benchmarks are 0%.

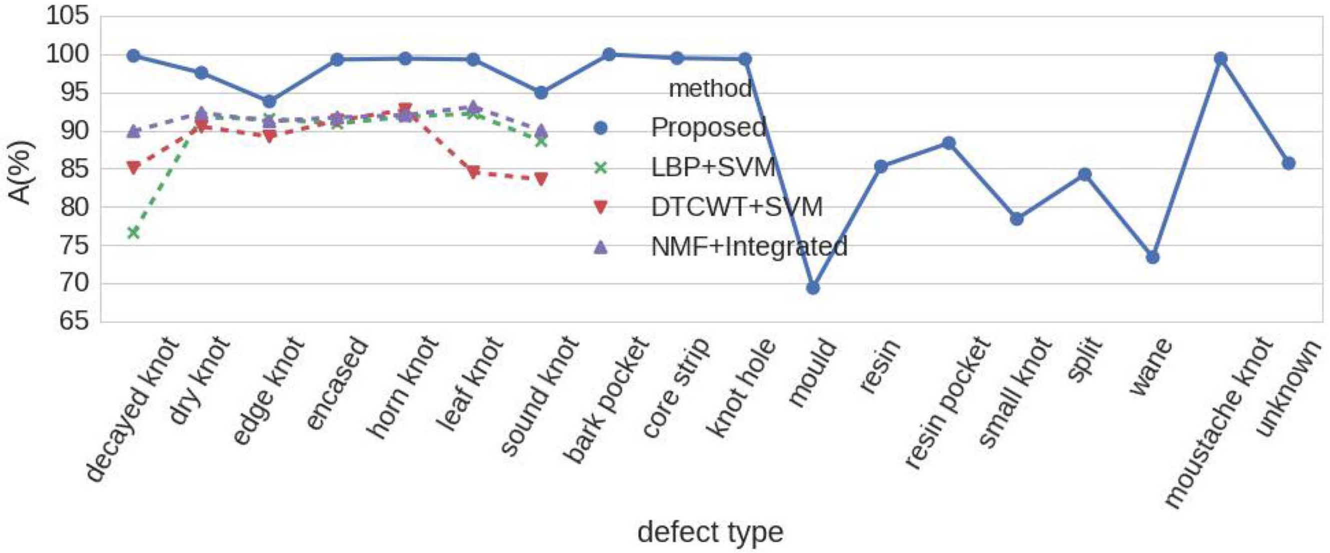


Fig. 14. Accuracies for wood defect detection using proposed method, LBP+SVM [36], DTCWT+SVM [36], and NMF+integrated [36]. Compared to the benchmark methods, proposed method improves accuracy of decayed knot, dry knot, edge knot, encased knot, horn knot, leaf knot, and sound knot by 9.86%, 5.23%, 2.29%, 7.57%, 6.70%, 6.18%, and 4.94%, respectively.

texture feature. An SVM classifier is trained on the features to distinguish defect and nondefect areas. Error escape (E) of [37] is reported against six types of defects. In [36], nonnegative matrix factorization (NMF) is applied on the images. Both the dual-tree complex wavelet transform (DTCWT) and LBP are tested for feature extraction. The misclassification (M) rate of [36] is reported on seven types of defects.

For the proposed method, performances on all the eighteen defect types are evaluated. The results are shown in Figs. 12–14. The defects that are not studied by [3], [36], and [37] are marked as missing data in the figures. From Fig. 12, the proposed method does not outperform the others in terms of the overall E, F, and M. However, the number of defect types included in evaluating the proposed method is larger than that of benchmark methods. Certain categories of defects that are not included in the benchmark methods contribute to large E and F except M. As such, comparing the performances in detecting each defect type is a fair criterion rather than the overall performance.

Figs. 13 and 14 show E and A of the proposed method and benchmark methods over each defect type. Fig. S-1 in the supplementary material includes examples of predictions produced by the algorithm. For these intermediate outputs, the heatmaps show the likelihood of a pixel being the anomaly, Otsu's thresholding turns the heatmaps into binary images, and Felzenswalb's segmentation removes noises and refines defect regions. From Fig. 13, error escape rate of proposed method is significantly lower than that of benchmark methods in terms of

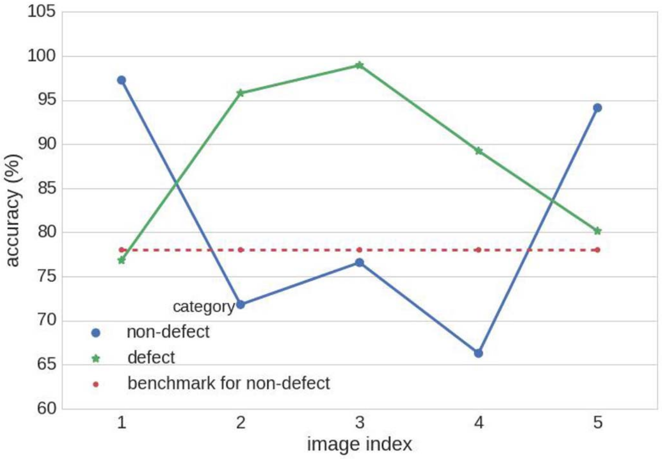


Fig. 15. Accuracy of binary classification for weld defect database using proposed method. The benchmark accuracy is based on results reported in [38], which is trained on a larger data set.

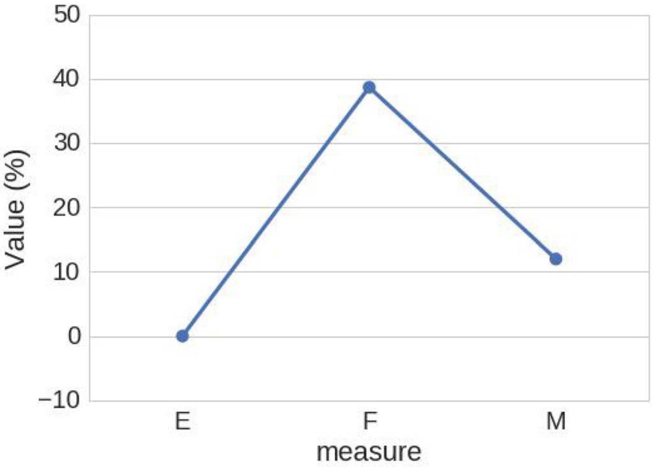


Fig. 16. Results of micro-structure dataset based on three criterion.

defect type *dry knot*, *resin*, and *resin pocket*. It is not the best-performing algorithm in terms of detecting *sound knot* and *split*. However, the performance is only slightly downgraded compared to the benchmark methods. For sound knot defect, E of proposed method is 3.13% compared to 1.5% of the best-performing benchmark. For *split* defect, it is 13.51% versus 13.3%. In Fig. 14, the proposed method outperforms three methods in [36] for all seven defect types.

2) Segmentation on Weld Defect Database: For the weld defect database, it is split into five training images and five testing images. For each training image, 100 defect and 100 nondefect patches are selected for training the patch classifier. Although patch size 25 × 25 is the one with highest classification accuracy, patch size 75 × 75 is selected in the defect segmentation task. The number of patches required by 25 × 25 is nine times higher than that of 75 × 75. As such, choosing 75 × 75 patches largely reduces computational cost without much loss of accuracy. As to the segmentation step, σ is chosen as 0.8 and k is chosen as 200 according to the defect size. Valavanis and Kosmopoulos [38] is selected as a benchmark method. It first segments defect area by performing Sauvola local thresholding [39] and Felzenswalb's segmentation [28]. Next, it performs classification of defect based on geometric features and intensity-based descriptors [38].

In [38], several types of defect are included for evaluation. However, the information of defect type is not provided for the weld defect dataset in our experiment. What is more, the dataset in [38] includes twenty four radiographs while there are only ten images in our experiment. As such, the proposed method is not directly comparable with [38]. However, the results in [38] show the accuracies for predicting defect and nondefect pixels, which implies the performance of proposed method. The results in [38] shows that the accuracy for classifying nondefect pixels is 78%. Fig. 15 presents the results using proposed method. The accuracies for binary classification of five images ranges from 66.27% to 97.26%. Although the dataset is small, the performance of proposed method is still comparable to the results reported in [38].

An example of prediction result using the proposed method is shown in Fig. S-3 in the supplementary material. It shows that high intensity areas in heat map are the defect areas. However, it is noisy due to the boundary of pipe. By applying Otsu's method, the high intensity pixels are set as 1 and the low intensity as 0. With the segmentation algorithm, the noisy and small areas are removed. As such, the final predictions are obtained.

3) Segmentation Results on Microstructured Dataset: The proposed method is also applied on the industrial microstructured dataset. Existing ASI methods for microstructured data usually apply on regular patterns [40], [41]. However, the texture of these images are stochastic. In the field of ASI, little literature can be found for microstructured images of titanium fan blade. As a result, no benchmark method is included for this dataset.

For these microstructured images, engineers are interested in highlighting the defect areas and false positive is preferred to false negatives. Relatively large patches (256 × 256) are chosen to speed up the detection, as precise pixel-wise classification is not required. This dataset is labeled similarly as wood defect subset 2. As such, the evaluation criterion (2)–(6) are reused. The results are shown in Fig. 16 and examples of detection are shown in Fig. S-2 in the supplementary material.

The results show that proposed method has zero error escape rate and slightly high false alarm rate. In fact, this property is desirable for microstructured defect detection. False negative is always much more costly than false positives, as undetected defects in fan blades may lead to serious safety issues. The experimental results show that the proposed method is a good candidate for ASI of this microstructured dataset.

From the experimental results of classification and segmentation tasks, the proposed method is efficient for small training data. By applying feature transferring, it reduces the cost of data collection and model training, as the proposed method does not require large amount of training data and only parameters in the classification layer are optimized instead of the whole deep CNN. In addition, the proposed method is also generic as the extracted features from Decaf are high-level representations of patches and dataset-independent.

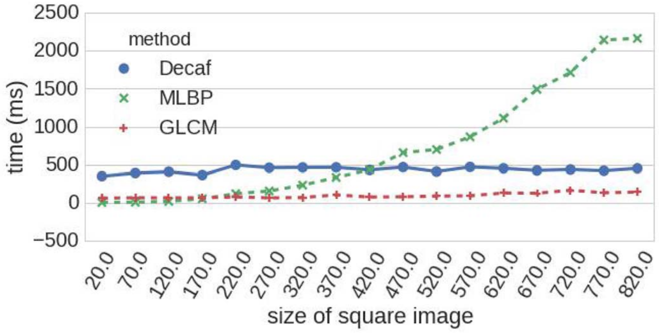


Fig. 17. Computational time for three feature extraction methods over different image sizes.

VI. Implementation Details

The proposed framework is implemented on a workstation with 24 cores and 32GB memory. The computational time of three feature extraction methods for patch size $x \times x$ is shown in Fig. 17. The computational cost of LBP-based method grows significantly as the patch size increase, due to its pixel-wise computation. The computational cost of transferred feature is independent of patch size, as Decaf always resizes input images to specific dimension. As the pretrained neural network is deep, the computational cost for feature extraction is higher than LBP and GLCM-based methods for small patches. In order to speed up the method, the feature extraction step is implemented in parallel. For defect segmentation of an image with 2000 patches, it takes 2 mins to generate heat maps on average.

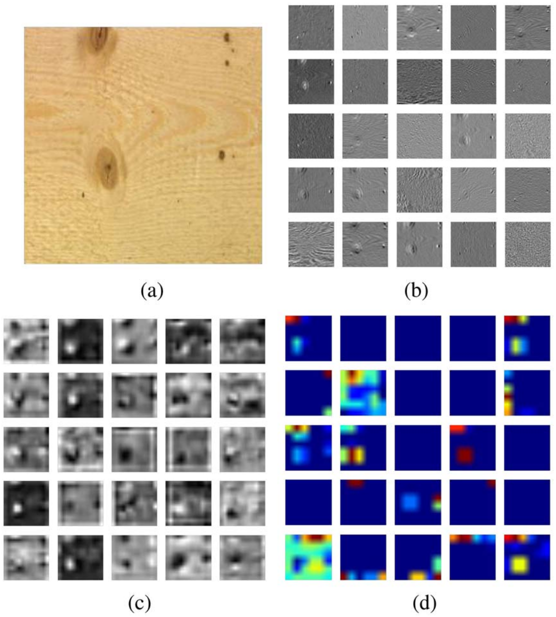
The algorithms are implemented using Python. The packages involved in this implementations are Scikit-Learn,[2] Numpy,[3] Scipy,[4] Scikit-Image,[5] and Decaf.[6] The pretrained weights for Decaf is publicly available.[7]

VII. Discussion

From the experimental results, the transferred feature outperforms hand-crafted features for all texture images. In addition, the proposed ASI method also performs well in defect segmentation. This section explores the reasons of such high performances.

A. Transferred Feature

From the classification results in Section V, transferred feature is a high-quality representation of surface inspection datasets. It is due to the complexity of object recognition task. Multiple types of information are essential in object recognition, such as texture and shape. CNN learns different levels of representation via propagation through its layers. An example of the type of information that Decaf captures from wood defects is shown in Fig. 18. The low-level layers detect edges and textures of input image, while the high level layers produce high-quality feature maps for defects. This example shows Decaf is highly adaptable to ASI domain.


(a) (b)

(c) (d)

Fig. 18. Visualization of features in different layers of Decaf for wood defect datasets. (a) Original image. (b) Feature map from first convolutional layer (Conv1). (c) Feature map from third convolutional layer (Conv3). (d) Feature map from fifth convolutional layer (Conv5).

B. Segmentation Method

As for the segmentation task, the results show that the proposed method has high performance and it is generic across various datasets. There are two major advantages of the proposed method. First, it can be trained on small dataset and yet can be highly accurate. For segmentation of weld defect, there are only five training images but the result of proposed method is still comparable to the state-of-art method. It shows the capability of the proposed scheme for small training data. It saves labor cost for industrial applications, since most visual inspection requires intensive manual work for collecting and labeling images. Second, the proposed method is easy to implement across datasets in various domains. Traditionally, ASI methods depend on the properties of defects, including size, texture, and colors. As a result, majority of existing ASI methods depend on hand-crafted features that can not be applied to large varieties of surfaces. The proposed framework is fully automatic and is robust to the changes in defect properties.

VIII. Conclusion

In this paper, a generic DL-based ASI method is presented. This method includes feature transferring from pretrained DL network and convolution of patch classifier over input image. The transferred feature significantly outperforms hand-crafted features in classification tasks. The improvement of accuracy ranges from 0.66% to 25.50% for three datasets. As for segmentation task, the proposed method reduces error escape rate

[2] http://scikit-learn.org/stable/
[3] http://www.numpy.org/
[4] http://www.scipy.org/
[5] http://scikit-image.org/
[6] https://github.com/UCB-ICSI-Vision-Group/decaf-release/
[7] http://daggerfs.com/

by 6.00%–19.00% in three defect types, improves accuracy by 2.29%–9.86% in seven defect types, and obtain 0.0% error escape rate for the microstructured dataset.

In the future, we will focus on two directions of research. One direction involves the speeding up the heat map generation process to enable real-time defect localization. The second direction is to automate the patch size selection process using machine learning methods.

References

[1] J. J. Liu and J. F. MacGregor, "Estimation and monitoring of product aesthetics: Application to manufacturing of 'engineered stone' countertops," *Mach. Vis. Appl.*, vol. 16, no. 6, pp. 374–383, 2006.

[2] V. Murino, M. Bicego, and I. A. Rossi, "Statistical classification of raw textile defects," in *Proc. IEEE Int. Conf. Pattern Recognit. (ICPR)*, vol. 4. Hong Kong, Aug. 2004, pp. 311–314.

[3] O. Silvén, M. Niskanen, and H. Kauppinen, "Wood inspection with non-supervised clustering," *Mach. Vis. Appl.*, vol. 13, nos. 5–6, pp. 275–285, 2003.

[4] F. Pernkopf, "Detection of surface defects on raw steel blocks using Bayesian network classifiers," *Pattern Anal. Appl.*, vol. 7, no. 3, pp. 333–342, 2004.

[5] X. Xie and M. Mirmehdi, "TEXEMS: Texture exemplars for defect detection on random textured surfaces," *IEEE Trans. Pattern Anal. Mach. Intell.*, vol. 29, no. 8, pp. 1454–1464, Aug. 2007.

[6] X. Xie, "A review of recent advances in surface defect detection using texture analysis techniques," *Electron. Lett. Comput. Vis. Image Anal.*, vol. 7, no. 3, pp. 1–25, 2008.

[7] J. Chen and A. K. Jain, "A structural approach to identify defects in textured images," in *Proc. IEEE Int. Conf. Syst. Man Cybern.*, vol. 1. Beijing, China, Aug. 1988, pp. 29–32.

[8] B. Mallik-Goswami and A. K. Datta, "Detecting defects in fabric with laser-based morphological image processing," *Text. Res. J.*, vol. 70, no. 9, pp. 758–762, 2000.

[9] W. Wen and A. Xia, "Verifying edges for visual inspection purposes," *Pattern Recognit. Lett.*, vol. 20, no. 3, pp. 315–328, 1999.

[10] J. Kittler, R. Marik, M. Mirmehdi, M. Petrou, and J. Song, "Detection of defects in colour texture surfaces," in *Proc. MVA*, Kawasaki, Japan, 1994, pp. 558–567.

[11] C.-W. Kim and A. J. Koivo, "Hierarchical classification of surface defects on dusty wood boards," *Pattern Recognit. Lett.*, vol. 15, no. 7, pp. 713–721, 1994.

[12] M. Niskanen, O. Silvén, and H. Kauppinen, "Color and texture based wood inspection with non-supervised clustering," in *Proc. Scandinavian Conf. Image Anal.*, Bergan, Norway, Jun. 2001, pp. 336–342.

[13] R. W. Conners, C. W. Mcmillin, K. Lin, and R. E. Vasquez-Espinosa, "Identifying and locating surface defects in wood: Part of an automated lumber processing system," *IEEE Trans. Pattern Anal. Mach. Intell.*, vol. PAMI-5, no. 6, pp. 573–583, Nov. 1983.

[14] F. Ade, N. Lins, and M. Unser, "Comparison of various filter sets for defect detection in textiles," in *Proc. Int. Conf. Pattern Recognit.*, vol. 1. Montreal, QC, Canada, Jul. 1984, pp. 428–431.

[15] S. A. H. Ravandi and K. Toriumi, "Fourier transform analysis of plain weave fabric appearance," *Text. Res. J.*, vol. 65, no. 11, pp. 676–683, 1995.

[16] J. G. Campbell and F. Murtagh, "Automatic visual inspection of woven textiles using a two-stage defect detector," *Opt. Eng.*, vol. 37, no. 9, pp. 2536–2542, 1998.

[17] J. Hu, H. Tang, K. C. Tan, and H. Li, "How the brain formulates memory: A spatio-temporal model research frontier," *IEEE Comput. Intell. Mag.*, vol. 11, no. 2, pp. 56–68, May 2016.

[18] A. Conci and C. B. Proença, "A fractal image analysis system for fabric inspection based on a box-counting method," *Comput. Netw. ISDN Syst.*, vol. 30, nos. 20–21, pp. 1887–1895, 1998.

[19] F. S. Cohen, Z. Fan, and S. Attali, "Automated inspection of textile fabrics using textural models," *IEEE Trans. Pattern Anal. Mach. Intell.*, vol. 13, no. 8, pp. 803–808, Aug. 1991.

[20] J. Schmidhuber, "Deep learning in neural networks: An overview," *Neural Netw.*, vol. 61, pp. 85–117, Jan. 2015.

[21] A. Krizhevsky, I. Sutskever, and G. E. Hinton, "Imagenet classification with deep convolutional neural networks," in *Proc. Adv. Neural Inf. Process. Syst.*, 2012, pp. 1097–1105.

[22] P. Sermanet *et al.*, "OverFeat: Integrated recognition, localization and detection using convolutional networks," in *Proc. Int. Conf. Learn. Represent. (ICLR)*, Banff, AB, Canada, Apr. 2014, pp. 1–16. [Online]. Available: https://arxiv.org/pdf/1312.6229v4.pdf

[23] J. Donahue *et al.*, "DeCAF: A deep convolutional activation feature for generic visual recognition," in *Proc. Int. Conf. Mach. Learn.*, Beijing, China, Jun. 2014, pp. 647–655.

[24] J. Yosinski, J. Clune, Y. Bengio, and H. Lipson, "How transferable are features in deep neural networks?" in *Proc. Adv. Neural Inf. Process. Syst.*, Montreal, QC, Canada, 2014, pp. 3320–3328.

[25] A. S. Razavian, H. Azizpour, J. Sullivan, and S. Carlsson, "CNN features off-the-shelf: An astounding baseline for recognition," in *Proc. IEEE Conf. Comput. Vis. Pattern Recognit. Workshops*, Columbus, OH, USA, 2014, pp. 806–813.

[26] M. Cimpoi, S. Maji, I. Kokkinos, S. Mohamed, and A. Vedaldi, "Describing textures in the wild," in *Proc. IEEE Comput. Vis. Pattern Recognit. (CVPR)*, Columbus, OH, USA, Jun. 2014, pp. 3606–3613.

[27] N. Otsu, "A threshold selection method from gray-level histograms," *Automatica*, vol. 11, nos. 285–296, pp. 23–27, 1975.

[28] P. F. Felzenszwalb and D. P. Huttenlocher, "Efficient graph-based image segmentation," *Int. J. Comput. Vis.*, vol. 59, no. 2, pp. 167–181, 2004.

[29] K. Song and Y. Yan, "A noise robust method based on completed local binary patterns for hot-rolled steel strip surface defects," *Appl. Surface Sci.*, vol. 285, pp. 858–864, Nov. 2013.

[30] D. Mery *et al.*, "GDXray: The database of X-ray images for non-destructive testing," *J. Nondestruct. Eval.*, vol. 34, no. 4, pp. 1–12, 2015.

[31] T. Ojala, M. Pietikäinen, and T. Mäenpää, "Multiresolution gray-scale and rotation invariant texture classification with local binary patterns," *IEEE Trans. Pattern Anal. Mach. Intell.*, vol. 24, no. 7, pp. 971–987, Jul. 2002.

[32] R. M. Haralick, K. Shanmugam, and I. H. Dinstein, "Textural features for image classification," *IEEE Trans. Syst., Man, Cybern.*, vol. SMC-3, no. 6, pp. 610–621, Nov. 1973.

[33] M. Nadler and E. P. Smith, *Pattern Recognition Engineering*. New York, NY, USA: Wiley, 1993.

[34] K. Song, S. Hu, and Y. Yan, "Automatic recognition of surface defects on hot-rolled steel strip using scattering convolution network," *J. Comput. Inf. Syst.*, vol. 10, no. 7, pp. 3049–3055, 2014.

[35] M. I. Chacon and G. R. Alonso, "Wood defects classification using a SOM/FFP approach with minimum dimension feature vector," in *Proc. Adv. Neural Netw. (ISNN)*, Chengdu, China, May 2006, pp. 1105–1110.

[36] S.-Y. Wu, Z. Zhang, and L. Feng, "Statistical feature representations for automatic wood defects recognition research and applications," in *Proc. IEEE Asia–Pac. Conf. Comput. Intell. Ind. Appl. (PACIIA)*, vol. 1. Wuhan, China, Nov. 2009, pp. 19–22.

[37] W. Song *et al.*, "Wood materials defects detection using image block percentile color histogram and eigenvector texture feature," in *Proc. 1st Int. Conf. Inf. Sci. Mach. Mater. Energy*, Chongqing, China, Apr. 2015, pp. 2022–2026.

[38] I. Valavanis and D. Kosmopoulos, "Multiclass defect detection and classification in weld radiographic images using geometric and texture features," *Expert Syst. Appl.*, vol. 37, no. 12, pp. 7606–7614, 2010.

[39] R. R. da Silva, M. H. S. Siqueira, M. P. V. de Souza, J. M. A. Rebello, and L. P. Calôba, "Estimated accuracy of classification of defects detected in welded joints by radiographic tests," *NDT E Int.*, vol. 38, no. 5, pp. 335–343, 2005.

[40] H. Jia, Y. L. Murphey, J. Shi, and T.-S. Chang, "An intelligent real-time vision system for surface defect detection," in *Proc. IEEE Int. Conf. Pattern Recognit. (ICPR)*, vol. 3. Hong Kong, Aug. 2004, pp. 239–242.

[41] K. J. Maayah, H. S. Gallarda, Jr., L. Srinivasan, R. Barnard, and J. Liu, "Automated repetitive array microstructure defect inspection," U.S. Patent 7 065 239, Jun. 20, 2006.

Ruoxu Ren received the B.Sc. (Hons.) and B.Soc.Sc. (Hons.) degrees from the National University of Singapore, Singapore, in 2013, where she is currently pursuing the Ph.D. degree with the Department of Electrical and Computer Engineering.

She is a Computational Engineering Technologist with Applied Technology Group, Rolls-Royce, Singapore. Her current research interests include deep learning, computer vision, natural language processing, and ASI for aerospace industry.

Terence Hung received the Ph.D. degree in electrical engineering from the University of Illinois at Urbana–Champaign, Champaign, IL, USA, in 1993.

He had held the position of the Deputy Executive Director, Institute of High Performance Computing, Singapore. He is currently the Chief of Computational Engineering, Rolls-Royce Singapore Pte. Ltd., Singapore. He leads a team of data scientists to develop and apply data analytics across Rolls-Royce's businesses and product lifecycle. He also manages the Data Analytics and Complex Systems Programme with the RR@NTU Corporate Lab, Nanyang Technological University, Singapore, and overseas long-term relationships with A*STAR, Singapore and universities. He is an Adjunct Associate Professor with the School of Computer Engineering, Nanyang Technological University. He assisted the Executive Director to oversee the operations and strategic directions of the institute. In an earlier appointment as the Director of the Computing Science Department, he led over 40 researchers in areas such as HPC, grid/cloud computing, data analytics, digital modeling, and visualization. He had also held the position of VP Technology with Commerce Exchange Pte. Ltd. in the area of e-commerce, e-financial, and e-logistics.

Dr. Hung serves in various professional bodies such as Microsoft's Technical Computing Executive Advisory Council, the Gerson Lehrman Group, HP-CAST Board and Singapore's National Cloud Computing Advisory Council.

Kay Chen Tan (SM'08–F'14) received the B.Eng. (First Class Hons.) degree in electronics and electrical engineering and the Ph.D. degree from the University of Glasgow, Glasgow, U.K., in 1994 and 1997, respectively.

He is a Full Professor with the Department of Computer Science, City University of Hong Kong, Hong Kong. He has published over 200 refereed articles and five books.

Dr. Tan is the Editor-in-Chief of the IEEE TRANSACTIONS ON EVOLUTIONARY COMPUTATION, was the Editor-in-Chief of the *IEEE Computational Intelligence Magazine* from 2010 to 2013, and currently serves as the Editorial Board Member of over 20 journals. He is an elected member of the IEEE CIS AdCom from 2017 to 2019 and is an IEEE CIS Distinguished Lecturer from 2015 to 2017.