Journal of Ambient Intelligence and Humanized Computing
https://doi.org/10.1007/s12652-021-03396-2

ORIGINAL RESEARCH

Detection and localization of defects on natural leather surfaces

Y. S. Gan[1] · Sze-Teng Liong[2] · Danna Zheng[3] · Yiyang Xia[3] · Shuli Wu[3] · Mengchen Lin[3] · Yen-Chang Huang[4]

Received: 13 January 2021 / Accepted: 8 July 2021

Abstract
Defects that appear on a leather surface may be the result of natural variations or poor handling during the manufacturing process. Visual inspection in the factory is one of the essential steps in the process of quality assurance. It should be done before the finished products are being dispatched to the customer. Thus far, the detection of the leather defects is still carried out manually, which is labour intensive, tedious, and might be liable to human error. Therefore, in this paper, we propose an automatic leather defect localization and detection system by employing a series of digital image processing methods based on deep learning. Succinctly, a convolutional neural network (CNN) is utilized to perform the detection task, that is to determine the presence of the defect on a leather patch. Then, the detected defective leather patch is processed to the localization operation, which is to identify the boundary coordinates in pixel level. For the detection task, the result achieved using AlexNet as the feature descriptor and SVM as the classifier is 100%. For the localization stage, we have demonstrated that the instance segmentation technique, Faster R-CNN outperforms the YOLOv2 by obtaining the Intersection over Union (IoU) of 73%. In addition, extensive experiments and comparisons of the state-of-the-art approaches are presented to verify the effectiveness of the proposed algorithms.

Keywords Leather · Defect · Detection · Localization · Faster R-CNN · YOLOv2

1 Introduction

Leather is made from the animal rawhide and skin, including cows, pigs, sheep, goats and crocodiles. It is a natural material that is water-resistant, flexible and durable. Due to many unexpected external or human factors, there will be some uneven patches and spots appearing on the surfaces on the animal skins. For instance, the common visible defects are wrinkles, cuts, insect bites, scabies and embossed marks left by the wound. Therefore, the leather pieces will be generally denatured through several chemical and physical refinements, such as the preparatory stages, tanning, crusting and surface coating. Nonetheless, there will be new kind of man-made defects created during the leather manufacturing due to improper processing. This type of issues contributing to downgrading the quality of the leather and thus affecting the sales price.

Leather can produce wide range of luxury goods, such as furniture, luggage, garments, decoration and others. Figure 1 shows the global revenue growth for the luxury leather goods from 2010 to 2019 (Statista 2019). This also reflects that the standard of living of people has been increased. The leather manufacturing companies include the brands like Louis Vuitton, Rimowa, Gucci, Coach, Michael Kors, Tumi and so on.

To accelerate the leather processing speed in the industry, this paper aims to focus on the visual inspection stage. Concretely, it is the process to monitor the quality of leather by identifying the defective area on the surface. To date, there are relatively few works, such as Liong et al. (2019a), Villar et al. (2011), Jian et al. (2010), Pistori et al. (2018), Kasi et al. (2014) had been carried out to perform the identification of

✉ Sze-Teng Liong
stliong@fcu.edu.tw

✉ Yen-Chang Huang
ychuang@mail.nutn.edu.tw

[1] School of Architecture, Feng Chia University, Taichung 40724, Taiwan

[2] Department of Electronic Engineering, Feng Chia University, Taichung, Taiwan

[3] School of Electrical and Computing Engineering, Xiamen University Malaysia, Jalan Sunsuria, Sepang, Selangor, Malaysia

[4] Department of Applied Mathematics, National University of Tainan, Tainan, Taiwan

Published online: 15 August 2021

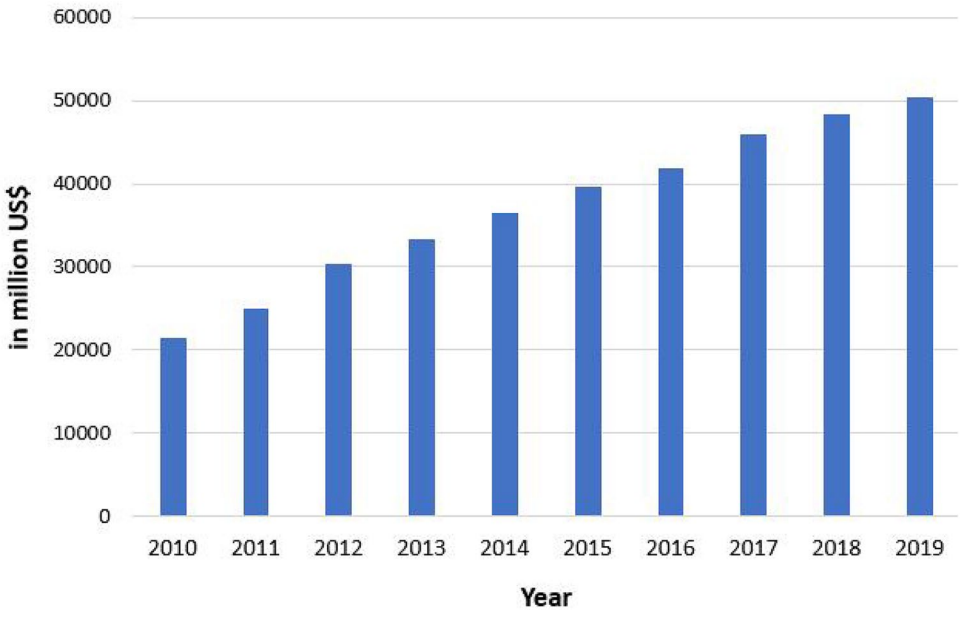

Fig. 1 Increasing revenue growth for the luxury leather goods from 2010 to 2019. Source: Luxury Leather (Statista 2019)

leather defect using automatic image processing techniques. Nowadays, most of the industries still employing the manual way, which requires human resources to sort out the defective part of the leather pieces. Such a manual based assortment is time-consuming, highly subjective, may be prone to error and thus lead to dispute during transactions. Therefore, it is essential to digitalize and vectorize the leather to allow the automatic grade classification and defect localization/ segmentation processes.

There are three primary objectives in this paper:

1. The leather patches are categorized as either defective or non-defective on the basis of its surface color by employing popular convolutional neural network and classifiers.
2. The boundaries of the defect regions with accurate XY coordinates are identified by adopting two types of pixel-wise segmentation networks.
3. Comprehensive quantitative and qualitative comparisons are presented for state-of-the-art methods. In addition, multivariate analysis is conducted and further investigations are derived for performance assessment.

Organization of the paper is as follows: Sect. 2 presents a detailed literature review on automatic detection and localization on leather. Section 3 elaborates the proposed leather defect identification system with detailed implementation description. Database used and experimental configurations are discussed in Sect. 4. The performance results are summarized in Sect. 5. Finally, Sect. 6 highlights the concluding remarks.

2 Related work

There are two major tasks in an automated leather defect identification system, i.e., detection and localization. The former classifies whether the leather patch contains defects or not. Meanwhile, the latter locates the positions of the defective regions. It should be noted that, until now, not many researchers work on the localization of leather defects. This section discusses prominent existing studies related to automated leather inspection system.

2.1 Detection

One of the earliest works that employs neural network in leather defect detection is Jian et al. (2010). They combine Feed-forward Neural Network (FNN) to select optimal attributes that can effectively describe the defects features on the leather surface. Then, a decision tree is adopted to classify the defects type. The defects types include knife hole, nicks, imprint injury and aberration. The overall detection accuracy is more than 90%. However, the total number of samples is considered few (i.e., 200 images). Thus, the experimental evidence is not sufficiently conclusive.

Similar detection task is carried out by Villar et al. (2011). They propose an automated system to classify the defect types (i.e., open cut, close cut and fly bite) on the wet blue leather. In brief, they utilizes Sequential Forward Selection (Jain et al. 2000) to extract the features from both the grayscale and the color model images. The overall detection results obtained were more than 95%.

Jawahar et al. (2014) classify the leather features based on the Wavelet Statistical Feature (WFS) and Wavelet Co-occurrence Feature (WCF) feature descriptors. Then, the detection is performed by adopting Support Vector Machine (SVM). Among 700 leather patches, 500 of them are defective and 200 has no defect. The data set is partitioned into the ratio of 7:3 for training and testing, respectively. The reported experimental results shows that the proposed algorithm can achieve high detection accuracy (i.e., 99%).

A 6-step defect inspection method is proposed by Bong et al. (2018) to categorize the type of the leather defects (i.e., scars, scratches and pinholes). An image capturing system is proposed, which consists of the camera and a light source. There are 2500 images being elicited from the setup. Among them, 2000 are the training data and 500 testing data. Each image will be first pass through a series of pre-processing methods, such as boundary detection, morphological operations and resizing. SVM is adopted as classifier to recognize the features extracted, like the color moments, color correlograms, Zernike moments, and texture. As a result, an average detection result of 98.8% is obtained.

Different from the conventional automatic leather detection analysis, Pistori et al. (2018) consider both the wet blue and raw hide leathers in their experiment. Particularly, they evaluate the proposed approach on four common defect types: tick marks, brand marks made from hot iron, cuts and scabies. The GLCM features are encoded from different input size of leather patches, then examine on several supervised learning classifiers [(SVM, Radial Basis Functions Networks (RBF) and Nearest Neighbours (KNN)]. It has been demonstrated that exploiting KNN with 40×40 spatial resolution attains a perfect detection rate of 100%. However, the dataset only has a total of 16 images, which may not provide convincing evidence to verify the robustness of the proposed method.

A similar classification work to Pistori et al. (2018) is presented by another group of research team (Jawahar et al. 2021) recently, in which the GCLM approach is exploited to extract the features of the leather defect patterns. Succinctly, a Fast Convergence Particle Swarm optimization (FCPSO) algorithm is introduced to effectively classify the leather patches into either the defective or non-defective groups. Then, several different types of classifiers such as neural network, Decision Tree, SVM, Naïve Bayes, KNN, and Random Forest (RF) are employed. As a result, the binary classification task achieves an accuracy of 88.64% when evaluated on a total of 200 images. Although the data sample has been increased by 12 times compared to Pistori et al. (2018), the algorithm may remain doubtable in such a small sample of studies.

On the other hand, one of the early research teams that adopted neural network on automatic leather fabric defect inspection is conducted by Kwak et al. (2000). A simple feedforward artificial neural network (ANN) (multilayer perceptron models) with one and two hidden layers (Ventura and Chen 1996) are constructed for the defect classification. Experiments show that two-layered perceptrons is more promising compared to that of three-layered when tested on 140 samples. Besides, they also demonstrate that the proposed method outperformed decision tree classifier.

Recently, Liong et al. (2019a) intend to identify the leather images that contain tick bite defects. Succinctly, Canny edge detection is employed to enhance the image quality. Next, artificial neural network (ANN) is adopted as the feature extractor and classifier. Among 2000 images, the highest result achieved is 80%. Since the dataset has imbalanced issue (i.e., more non-defective images), the authors tested the algorithm by reconstructing the dataset to form 1:1, 1:2, 1:3 data distributions for defective:non-defective images. They have shown that the detection results for 1:3 data subset is higher compared to those 1:1 and 1:2.

On the other hand, Winiarti et al. (2018) attempted to distinguish among five leather types, namely, monitor lizard, crocodile, sheep, goat and cow skin. The feature descriptor used is the pre-trained CNN method, viz, AlexNet (Krizhevsky et al. 2012). This paper is capable of generating an overall detection accuracy of 99.9%. The experimental results also show that the pre-trained network can better extract the discriminative features compared to the handcrafted features (i.e., statistical texture measurement of GLCM, color moments and Gray Level co-occurrence characteristics). Note that, they do not perform any leather defect detection.

More comprehensive analysis regarding the effectiveness of distinct deep learning networks to different types of the leather defect (i.e., scratches, rotten surfaces, holes, and needle eyes) is reported in Deng et al. (2020). Particularly, the neural network architectures investigated include LeNet5 (LeCun et al. 1998), ResNet50 (He et al. 2016), Faster R-CNN (Ren et al. 2016), etc. However, despite promising accuracy is yielded (i.e., average accuracy is $\sim 95\%$), the algorithm is designed such that the network is designated in dealing with one type of defect each time. On the other hand, Aslam et al. (2020) suggest performing the transfer learning technique after extracting the features from the datasets such as HAM10000 (Tschandl et al. 2018) (contains 10000 pigmented skin lesions images that made up from 7 classes), CODEBRIM (Mundt et al. 2019) (contains 8300 concrete defect images that made up from 5 classes) and ImageNet (Deng et al. 2009) (contains 1.2 million images that made up from 1000 classes) datasets. Nonetheless, the highest accuracy yielded in the binary classification task is 83%. This may be ascribed to the class imbalance problem, as the total number of defective images is about 2.5 times more than the non-defective ones.

2.2 Localization

Serafim (1992) utilizes Laplacian pyramids analysis from the auto regressive models. Both the 0-level and 1-level qualitative results are reported to demonstrate the wrinkle defects localization. Since this is one of the earlier works (in 1992) performing localization on leather, it is not clear that the proposed method evaluated on how many leather images. In addition, there is no measurement metric to quantify the localization performance.

Similar to the research work carried out in Serafim (1992), Kasi et al. (2014) suggest an automatic adaptive edge detection algorithm to identify the defects in the leather surface. First, the image leather is passed through a Gaussian filter to eliminate the noise. Then, a Sobel operator is applied to detect the defect edges. A thresholding technique is performed to normalize the image intensity. However, the result obtained by this method had not been quantified and thus may not be comparable to other methods.

Liong et al. (2019b) propose a fully automated defect detection and marking system on calf leather. They first labeled the ground-truth (defects) manually on each of the images collected using a robot arm. Then, the annotated images are served as the input data to train the Mask R-CNN instance segmentation architecture. This trained architecture is used to perform the defect identification on the test images. Furthermore, they modify the LBP method to sort out the a few boundary points on the defective areas. The segmentation accuracies yielded are 91.50% and 70.35% on the train and test datasets, respectively.

Based on the above-described review, it can be summarized that most analyses utilized hand-crafted methods. In this paper, we aim to distinguish our work from others by applying state-of-the-art CNNs to establish a complete, automated leather defect detection and localization system. To the best of our knowledge, this is the first attempt that is comprised of the two-stage end-to-end model.

3 Method proposed

The overall detection and localization processes are illustrated in Fig. 2. Specifically, the proposed image processing system incorporates two stages: (a) detection—to categorize the images into either defective or non-defective classes. The feature extractor adopted is AlexNet. Then, two classifier tools are employed, viz., SVM and softmax; (b) localization—the detected defective images are further processed to identify the boundary of the defect region. Two efficient instance-level segmentation networks are utilized to achieve the defect localization, namely, YOLOv2 and Faster R-CNN. Detailed technical description of the system are elaborated in Sects. 3.1 and 3.2 below.

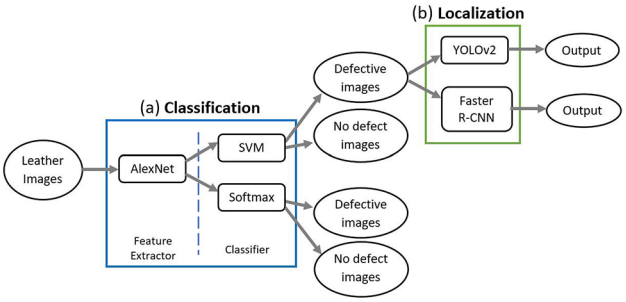

Fig. 2 Flowchart of the detection and localization processes

Table 1 Modified AlexNet configuration for the convolutional (*Conv*) layers, pooling (*Pool*) layers, fully connected (*FC*) layer and output sofmax layer

Layer	Filter size	# Filters	Stride	Padding	Output size
Input	–	–	–	–	$227 \times 227 \times 3$
Conv 1	$11 \times 11 \times 3$	96	[4, 4]	[0, 0, 0, 0]	$55 \times 55 \times 96$
Pool 1	3×3	–	[2, 2]	[0, 0, 0, 0]	$27 \times 27 \times 96$
Conv 2	$5 \times 5 \times 48$	256	[1, 1]	[2, 2, 2, 2]	$27 \times 27 \times 256$
Pool 2	3×3	–	[2, 2]	[0, 0, 0, 0]	$13 \times 13 \times 256$
Conv 3	$3 \times 3 \times 256$	384	[1, 1]	[1, 1, 1, 1]	$13 \times 13 \times 384$
Conv 4	$3 \times 3 \times 192$	384	[1, 1]	[1, 1, 1, 1]	$13 \times 13 \times 384$
Conv 5	$3 \times 3 \times 192$	256	[1, 1]	[1, 1, 1, 1]	$13 \times 13 \times 256$
Pool 5	3×3	–	[2, 2]	[0, 0, 0, 0]	$6 \times 6 \times 256$
FC 6	–	–	–	–	$1 \times 1 \times 4096$
FC 7	–	–	–	–	$1 \times 1 \times 4096$
FC 8	–	–	–	–	$1 \times 1 \times 4096$
Softmax	–	–	–	–	2×1

3.1 Defect detection

Amongst plenty of pre-trained CNN choices, such as AlexNet (Krizhevsky et al. 2012), GoogLeNet (Szegedy et al. 2015), SqueezeNet (Iandola et al. 2016), ShuffleNet (Zhang et al. 2018), DenseNet-201 (Huang et al. 2017) and VGG-16 (Simonyan and Zisserman 2014), we opt for AlexNet. This is due to its simple and effective architecture, which contains all the fundamental cascaded stages, such as convolution layers, pooling layers, rectified linear unit (ReLU) layers and fully connected layers. The basic structure of AlexNet is shown in Table 1, which consists of five convolution layers, three pooling layers and three fully connected layers. AlexNet is a classification model which was initially trained on more than a million images from the ImageNet (Deng et al. 2009) dataset, which contains 1000 categories. The transfer learning technique adopted herein is to overcome the small data regime issue faced in the experiment.

To perform transfer learning of this pre-trained network into our experiment, we performed slight modifications on some of the parameters, particularly the last fully connected layer. Note that the classifier type of the original AlexNet is softmax. Additionally, we employed the SVM classifier to evaluate the features extracted from the network. Therefore, there are two detection results obtained: (a) AlexNet + softmax, and; (b) AlexNet + SVM.

The softmax classifier that uses the cross-entropy loss function in AlexNet with L2 regularization, is defined as:

$$L = -\frac{1}{N}\left[\sum_{i=1}^{C} y_i \log\left(\frac{e^{z_i}}{\sum_{j=1}^{N} e^{z_j}}\right)\right] + \lambda \sum_{k=1}^{m} (w^{[k]})^2, \tag{1}$$

where y_i is the one-hot label of the classes of input N, N is the number of input samples, C is number of classes, $w^{[k]}$ is the vector of weights at layer $k-th$, $m \in \mathbb{Z}^+$ is the number of layers and z_i is the output vectors of model at $i-th$ sample defined as

$$z_i = f_{sigmoid}(f_{ReLU}(X_i^m; W)), \tag{2}$$

such that

$$f_{ReLU}(x_i^m; W) = max\left(0, \Sigma_j^n X_j^{m-1} w_{ji}^{m-1} + b_i^{m-1}\right), \tag{3}$$

$$X_j^m = f_{ReLU}(X_j^m; W), \text{ and} \tag{4}$$

$$X_j^1 = \sum_k w_{kj}^0 X_k^0 + b_j^0. \tag{5}$$

The SVM classifier for AlexNet with L2 regularization is molded to:

$$L = \frac{1}{N}\sum_{i}^{N}\sum_{j\neq i}^{N}[max(0, f_{sigmoid}(f_{ReLU}(X_i^m; W)) - f_{sigmoid}(f_{ReLU}(X_j^m; W)) + \delta] + \lambda \sum w^2, \tag{6}$$

where X_i^m is similar to that of Eq. (4).

3.2 Defect localization

According to the designed system illustrated in Fig. 2, the defective images will be further processed to localize the defective region. The methods involved in this localization stage are Faster R-CNN and YOLOv2:

1. Faster R-CNN It is the combination of Fast R-CNN (Girshick 2015) and Region Proposal Network (RPN) (Ren et al. 2015). The brief structure of Faster R-CNN is shown in Fig. 3. Basically, Faster R-CNN uses convolution layers for feature extraction and redundancy reduction, then computes good candidate anchor boxes by using RPN. These features with differently-sized anchor boxes are passed to the ROI pooling layer to create equal proposal feature maps size. RPN has two main tasks to perform: bounding box regression and object detection. The bounding box regression adjusts the anchor boxes' coordinates while non-maximum suppression (NMS) operates on the anchor boxes according to the scores of object detection to resolve for accurate and precise boxes. The pseudocode of the NMS algorithm is listed in Algorithm 1. We employ pretrained

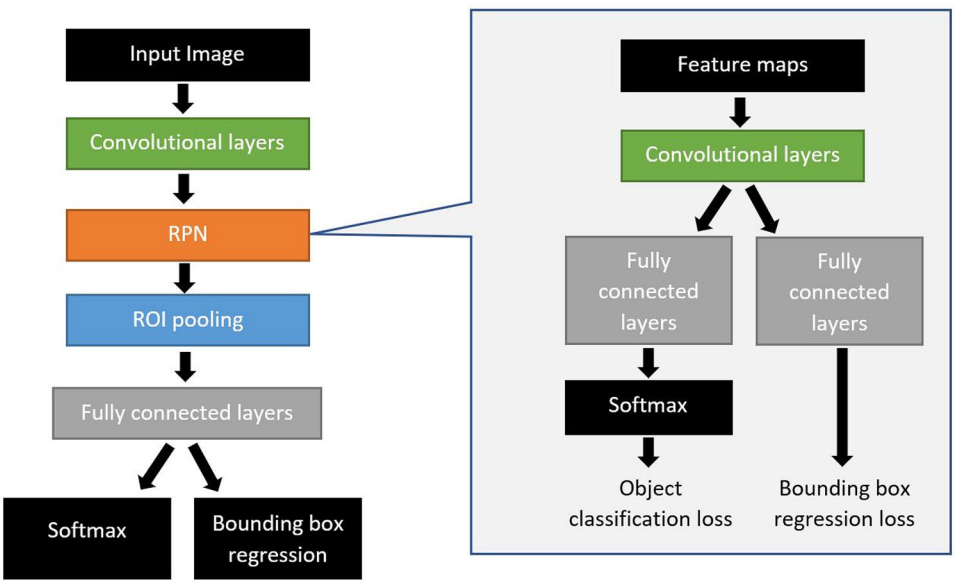

Fig. 3 Structure of Faster R-CNN

ResNet50 (He et al. 2016) as the basic network, as its ensembling subsets of residual modules enable effective iterative refinement by training through deeper networks. In brief, Faster R-CNN comprises two networks that train in two phases individually, i.e., RPN and Object Detection Network (ODN). The loss function of the RPN is defined as follows:

$$L_{loc}(p_i, t_i) = \frac{1}{N_{class}} \sum_i L_{class}(p_i, p_i^*) + \lambda \frac{1}{N_{reg}} \sum p_i^* L_{reg}(t_i, t_i^*), \tag{7}$$

where p_i is predicted probability of being an object for anchor i. t_i refers to the coordinates of the predicted bounding box for anchor i. L_{cls} is the log loss. p_i^* is the ground-truth label, L_{reg} is the smooth $L1$ loss, and t_i^* is the ground-truth box coordinates. Moreover, the regressor of the bounding box exploits the $L1$ loss on the position (x, y) of the top-left, the logarithm of the height and width of the box. In short, the loss is given as follows:

$$L(p, c_t, b_p, b_t) = L_{cls}(p, c_t) + \lambda[c_t \geq 1] L1_{loc}(b_t, b_p) \tag{8}$$

where p the predicted class scores, c_t the ground-truth of the class score, b_p is the predicted box coordinates and b_t is the ground-truth box coordinates. A 4-step alternating training algorithm is applied as follows:

(a) The network is adopted to initialize weights and train the RPN;
(b) Faster R-CNN is trained according to the region proposal from step (a) and the weights are updated;
(c) RPN is trained;
(d) Faster R-CNN is trained according to the region proposal from step (c).

2. YOLOv2 It is an improved model from YOLOv1, with lesser minimize localization errors and higher mean average precision. It is the state-of-the-art object detector that was performed on the databases like PASCAL VOC (Everingham et al. 2010) and COCO (Lin et al. 2014). YOLOv2 can adapt easily to different input image size and lead to good results by the multi-scale training. YOLOv2 outperforms YOLOv1 because the spatial resolution in the detection network is modified from 224×224 to 448×448. Besides that, a global average pooling operation is employed for better prediction. Furthermore, YOLOv2 uses the k-means clustering method to produce anchor boxes with higher resolution and smaller bounding boxes by calculating the distance function:

$$d(box, centroid) = 1 - \text{IoU}(box, centroid). \tag{9}$$

The location of the anchor box prediction is implemented to produce a better prediction location t_u. The bounding box locator for YOLOv2 is trained by using the localization loss defined in the following equation:

$$\lambda_{coord} \sum_{i=0}^{S^2} \sum_{j=0}^{B} \mathbb{1}_{ij}^{obj} \left[(x_i - \bar{x}_i)^2 + (y_i - \bar{y}_i)^2\right] + \lambda_{coord} \sum_{i=0}^{S^2} \sum_{j=0}^{B} \mathbb{1}_{ij}^{obj} \left[(\sqrt{w_i} - \sqrt{\bar{w}_i})^2 + (\sqrt{h_i} - \sqrt{\bar{h}_i})^2\right], \tag{10}$$

where $\mathbb{1}_{ij}^{obj}$ is the j_{th} bounding box with the appearance of the object in cell i. (x, y, w, h) denote the predicted coordinates of the top-left corner, width and height of the bounding box. In short, we propose a solution to solve the localization of leather defect. By using the AlexNet + softmax, the predicted location is expressed as:

$$loc = \begin{cases} t_u & argmax(z_0, z_1) = 1 \\ nil & argmax(z_0, z_1) = 0 \end{cases} \tag{11}$$

where $z_i = f_{sigmoid}(f_{ReLU}(x_i^m; W))$ and t_u is the bounding boxes predicted by YOLOv2/ Faster R-CNN On the other hand, using AlexNet + SVM (Table 2), the predicted location is formulated as:

$$loc = \begin{cases} t_u & z_0 - z_1 \geq 1, \forall i \neq j, \\ nil & z_0 - z_1 < 1, \forall i \neq j. \end{cases} \tag{12}$$

Table 2 The basic YOLOv2 Network

Layer	Operation	Filter/ pool size	Number of filter	Stride	Padding	Channel/ element	Anchor	Output size
Input	–	–	–	–	–	–	–	128×128×3
1	Convolution 1	3×3×3	16	[1 1]	[1 1 1 1]	-	-	128×128×16
	Normalization 1	–	–	–	–	16	–	128×128×16
	ReLU 1	–	–	–	–	–	–	128×128×16
	Pooling 1	2x2	–	[2 2]	[0 0 0 0]	–	–	64×64×16
2	Convolution 2	3×3×16	32	[1 1]	[1 1 1 1]	–	–	64×64×32
	Normalization 2	–	–	–	–	32	–	64×64×32
	ReLU 2	–	–	–	–	–	–	64×64×32
	Pooling 2	2×2	–	[2 2]	[0 0 0 0]	–	–	32×32×32
3	Convolution 3	3×3×32	64	[1 1]	[1 1 1 1]	–	–	32×32×64
	Normalization 3	–	–	–	–	64	–	32×32×64
	ReLU 3	–	–	–	–	–	–	32×32×64
	Pooling 3	2×2	–	[2 2]	[0 0 0 0]	–	–	16×16×64
4	Convolution 4	3×3×64	128	[1 1]	[1 1 1 1]	–	–	16×16×128
	Normalization 4	–	–	–	–	128	–	16×16×128
	ReLU 4	–	–	–	–	–	–	16×16×128
5	Convolution 5	3×3×128	128	[1 1]	Same	–	–	16×16×128
	Normalization 5	–	–	–	–	128	–	16×16×128
	ReLU 5	–	–	–	–	–	–	16×16×128
6	Convolution 6	3×3×128	128	[1 1]	Same	–	–	16×16×128
	Normalization 6	–	–	–	–	128	–	16×16×128
	ReLU 6	–	–	–	–	–	–	16×16×128
7	Convolution 7	1×1×128	24	[1 1]	[0 0 0 0]	–	–	16×16×24
	Transform	–	–	–	–	–	4	16×16×24
	Output	–	–	–	–	–	4	–

Algorithm 1 Non-Maximal Suppression (NMS)

$B \leftarrow$ List of detection boxes
$S \leftarrow$ Score of boxes
$t \leftarrow$ Threshold
$D \leftarrow \{\}$
repeat
  Find the detection box $B^{(i)}$ in B with the largest score $S^{(i)}$
  Assign $B^{(i)}$ to D
  repeat
    IF IoU(B^i, B^j) is greater than t
      Remove $B^{(j)}$ from B
      Remove $S^{(i)}$ from S
    END
  until All elements in B have been traversed
until B is empty

4 Experiment setup

4.1 Database

Our database contains 560 leather images with spatial resolution of 140 × 140×3. Among them, 280 images have noticeable open cut defects on the surface, while 280 images do not have defects at all. For the detection task (i.e., to distinguish between defective and non-defective images), three experiments were carried out with the ratio of train data to test data being 0.4, 0.6, and 0.8 respectively. The details of the database are shown in Table 3.

Table 3 Database configuration for detection and localization tasks

Experiment	Train set		Test set		Total
	Defect	Non-defect	Defect	Non-defect	
Detection—[0.4 train/ test split]	112	112	168	168	560
Detection—[0.6 train/ test split]	168	168	112	112	560
Detection—[0.8 train/ test split]	224	224	56	56	560
Localization	200	0	80	80	360

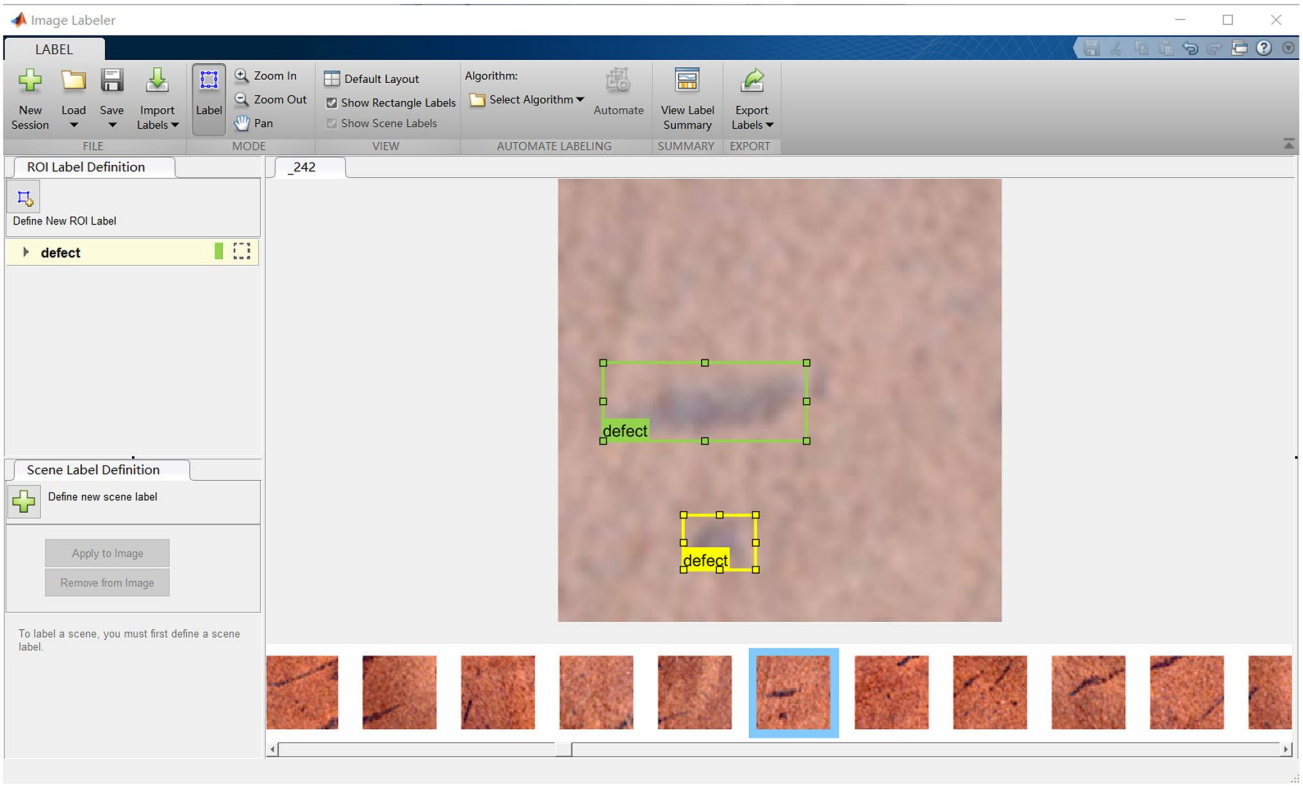

Fig. 4 Image Labeler to annotate the ground truth of the regions of defect

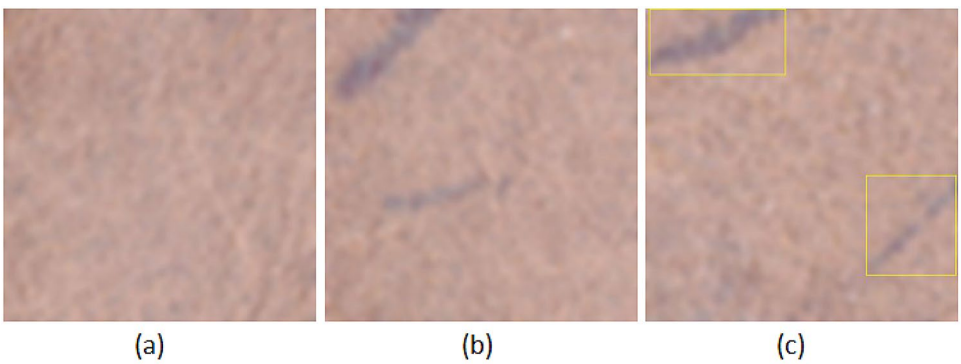

Fig. 5 Dataset sample: **a** no defect; **b** defective; **c** defective with rectangular region of interest (ROI) labels

Table 4 Training options of YOLOv2 and Faster R-CNN models

	YOLOv2	Faster R-CNN
Solver	Adam	Adam
Initial learn rate	0.0005	0.0001
MiniBatchSize	[2, 4, 8, 16, 32]	4
MaxEpochs	[30, 50, 100, 150]	15
Shuffle	Every epoch	Every epoch

As for the localization (i.e., to identify the defective regions), 200 defective images serve as the train data, whereas the remaining 80 defective images and 80 non-defective images are treated as the test data. The database details are stated in Table 3. Prior to the localization stage, the ground truth of the defect regions are required to be manually labeled to train the object detector. We opt for the MATLAB built-in object detector tool, Image Labeler. The screenshot of the Image Labeler application with marked rectangular region of interest (ROI) labels is illustrated in Fig. 4. This application allows the user to annotate the ROI labels, polyline ROI labels, pixel ROI labels and scene labels in an image. In our experiment, a rectangular label is chosen as the ROI. The steps to operate this application are described as follows:

1. An image is loaded from the local server.
2. A new ROI label is created and defined as "defect".
3. A rectangle bounding box is drawn around each defective area and it is regarded as the ROI.
4. The marked subjects are saved as the ground truth objects and used to train the YOLOv2 and Faster R-CNN architectures.

Figure 5 shows the samples of defect images, non defect images and labeled images.

The dataset is elicited using a robot arm DRV70L from Delta equipped with a Canon 77D camera fitted with a 135mm focal length lens. The robot arm moves in both the vertical and horizontal directions to capture the entire piece of leather surface. In addition, DOF D1296 Ultra High Power LED light is used to reduce the lighting flicker and any environmental interference.

4.2 Experiment configuration

The experiments are conducted using MATLAB 2019a on an Intel Core i7-6560U 2.20 GHz processor, RAM 8GB. All the input images are first resized to $227 \times 227 \times 3$ from the original spatial resolution of $140 \times 140 \times 3$. This is because in the classification, the AlexNet was trained with RGB values with the training data having the input size of $227 \times 227 \times 3$. On the other hand, for the localization stage, as there is no restriction on the input size, the input images of both the YOLOv2 and Faster R-CNN are resized back to the original size of $140 \times 140 \times 3$. This can also reduce the computational complexity and hence improve the execution speed. The training options for YOLOv2 and Faster R-CNN models are listed in Table 4.

4.3 Performance metrics

The experimental results report different types of evaluation metrics for leather defect detection and localization tasks. They are accuracy, F1-score and Intersection over Union (IoU). The metrics to evaluate the detection performance are accuracy and F1-score. Conceptually, accuracy is the number of correct predictions in the entire test set. F1-score indicates the harmonic average of precision (i.e., the rate of correctly predicted defective samples in all samples predicted as defects) and recall (i.e., the rate of correctly predicted defective samples in all actual defective samples).

$$\text{Accuracy} = \frac{\text{TP} + \text{TN}}{\text{TP} + \text{FN} + \text{TN} + \text{FP}}; \tag{13}$$

and

$$\text{F1-Score} = 2 \times \frac{\text{Precision} \times \text{Recall}}{\text{Precision} + \text{Recall}}, \tag{14}$$

for

$$\text{Precision} = \frac{\text{TP}}{\text{TP} + \text{FP}}, \tag{15}$$

and

$$\text{Recall} = \frac{\text{TP}}{\text{TP} + \text{FN}}, \tag{16}$$

where TP, TN, FN, and FP are the true positive, true negative, false negative and false positive, respectively. These four terms are defined as follows:

– TP: The number of defective leather images that the model predicts correctly.

Table 5 Comparison of detection accuracy [*Acc* (%)] and F1-score [(*F*1 (%)] using AlexNet + softmax and AlexNet + SVM

Train/test split	AlexNet + Softmax		AlexNet + SVM	
	Acc	F1	Acc	F1
0.4	97.02	97.08	**100**	**100**
0.6	**98.66**	**98.68**	99.55	99.56
0.8	97.32	97.33	**100**	**100**

Bold values represents the highest score

– TN: The number of non-defective leather images that the model correctly predicts.
– FP: The number of leather images that the model predicts there is a defect exists, but in fact there is none.
– FN: The number of leather images that the model wrongly classifies as non-defective leather, while in fact there is a defect.

On the other hand, for localization task, the metric used is IoU. It indicates the ratio of the overlap of ground truth with the detection result. The segmented defects are the output bounding boxes. Practically, there may exist more than one bounding box for each leather image, since an image may have multiple defective areas (as shown in Figure 5b). For each image, all the overlapping ratios will be computed. Then, the average IoU for the whole test set is reported as the final result. The IoU is expressed as:

$$\text{IoU} = \frac{\text{Detection Result} \bigcap \text{Ground Truth}}{\text{Detection Result} \bigcup \text{Ground Truth}} \tag{17}$$

5 Results and discussion

This section presents the experimental results for both the detection and localization with detailed analysis and discussion.

5.1 Detection performance

The detection result by adopting AlexNet + softmax and AlexNet + SVM with different division of train/test split is tabulated in Table 5. It can be seen that both the detection accuracy and F1-score reach 100%, when the train/test split are to 0.4 and

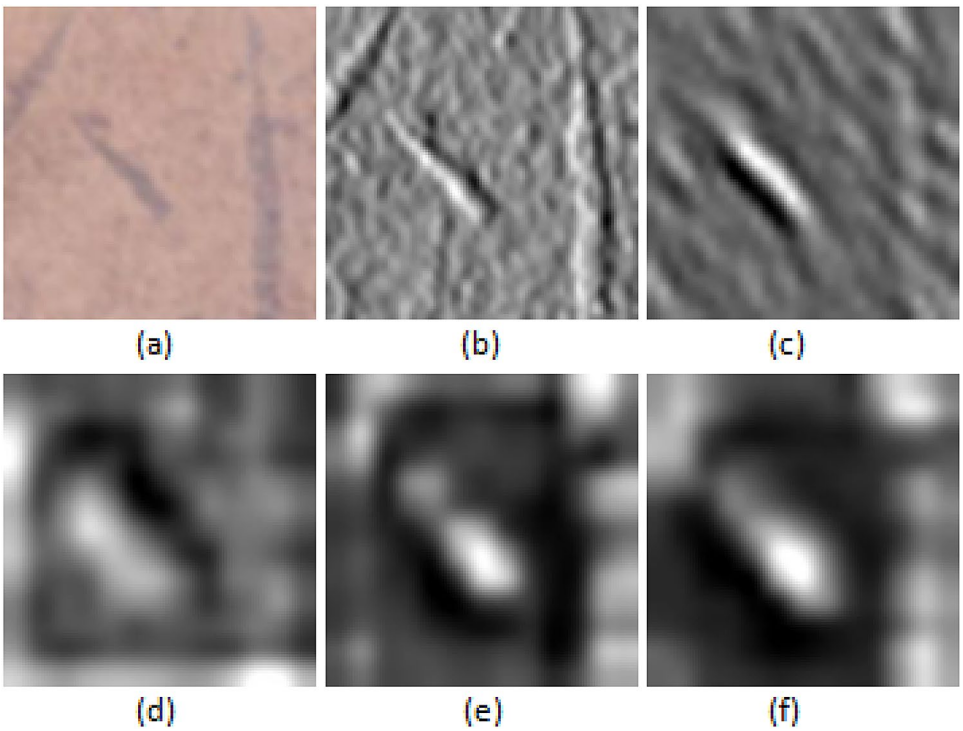

Fig. 6 Example of the activation feature maps from the AlexNet convolution layers. **a** Original image; **b** "conv1"; **c** "conv2"; **d** "conv3"; **e** "conv4", and; **f** "conv5"

Table 6 Comparison of detection accuracy to the state-of-the arts

No.	Detection task	Method	No. of sample	Acc (%)
1	**Defect vs no defect**	**Proposed method**	**560**	**100**
2	Defect vs no defect	ANN (Liong et al. 2019a)	2378	82.49
3	Defect vs no defect	(WSF+WCF) + SVM (Jawahar et al. 2014)	700	99.75
4	Multi-Defect type	SFS + SVM (Villar et al. 2011)	1769	96.48
5	Multi-Defect type	GLCM + KNN (Pistori et al. 2018)	16	100
6	Multi-Defect type	FNN + DT (Jian et al. 2010)	200	94
7	Multi-Defect type	Statistical features + SVM (Bong et al. 2018)	2500	98.8
8	Multi-Leather type	(GLCM+CNN) + SVM (Winiarti et al. 2018)	4157	99.97

Bold values represents the highest score

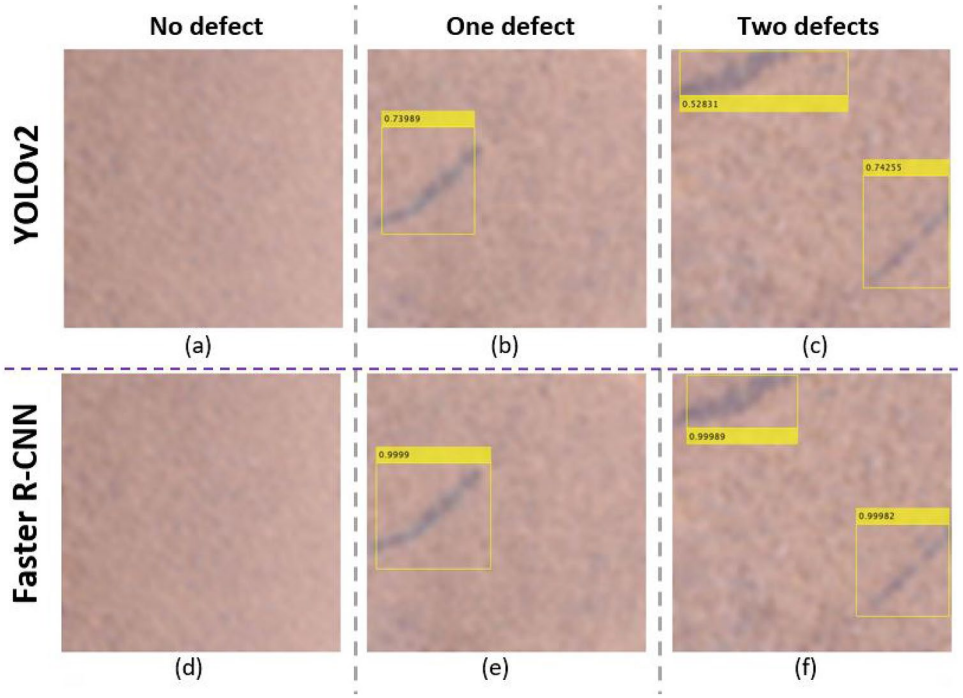

Fig. 7 Comparison of the results of **a**–**c** YOLOv2 and **d**–**f** Faster R-CNN

0.8 for AlexNet + SVM protocol. Notably, AlexNet + SVM obtains promising result when the size of the training dataset is small (i.e., 40% of data are the training set). Besides, all the results in AlexNet + SVM perform better than that of AlexNet + softmax. Nevertheless, AlexNet + softmax can reach significantly promising results (i.e., 98.66%), in 0.6 train/test split.

To visualize the attributes learned by AlexNet, we investigate the activation feature maps after passing through each convolution layer. Fig. 6 shows the example of the channels output from the first to the fifth convolution layer. It can be observed that there are obvious lines/ boundaries detected in "conv1" (i.e., Fig. 6b),

Table 7 Performance evaluation result using YOLOv2 (%): average IoU (*AvgIoU*), accuracy (*Acc*), recall (*Rec*), precision (*Pre*) and F1-score (*F*1)

Mini batch size	Epoch	Avg IoU	Acc	Rec	Pre	F1
2	30	65.85	97.50	97.50	97.50	97.50
	50	64.65	97.50	96.25	98.72	97.47
	100	61.61	93.75	91.25	96.05	93.59
	150	59.64	91.88	87.50	95.89	91.50
4	30	60.49	95.00	98.75	91.86	95.18
	50	64.77	95.63	93.75	97.40	95.54
	100	58.11	90.00	93.75	87.21	90.36
	150	53.75	89.38	86.25	92.00	89.03
8	30	64.69	**98.13**	**100**	96.39	**98.16**
	50	**69.24**	96.88	98.75	95.18	96.93
	100	59.88	91.25	93.75	89.29	91.46
	150	51.14	86.25	81.25	90.28	85.53
16	30	58.13	92.50	93.75	91.46	92.59
	50	68.11	93.75	96.25	91.67	93.90
	100	64.65	91.36	92.68	90.48	91.57
	150	64.65	89.82	92.86	87.64	90.17
32	30	27.94	75.00	51.25	**97.62**	67.21
	50	60.86	93.75	93.75	93.75	93.75
	100	64.65	92.02	92.77	91.67	92.22
	150	51.43	89.38	83.75	94.37	88.74

Bold values represents the highest score

Table 8 Confusion matrices using YOLOv2 method (mini-batch size = 8, epoch = 30)

		Predicted	
		Defective	Non-defective
Actual	Defective	**80**	0
	Non-defective	3	**77**

Bold values represents the highest score

Table 9 Best performance result of Faster R-CNN vs best performances results of YOLOv2 (%)

Method	Avg IoU	Acc	Rec	Pre	F1
Faster R-CNN	**73.47**	96.88	**100**	94.12	96.97
YOLOv2-1	69.24	96.88	98.75	95.18	96.93
YOLOv2-2	64.69	**98.13**	**100**	**96.39**	**98.16**

Bold values represents the highest score

Table 10 Confusion matrices using Faster R-CNN method

		Predicted	
		Defective	Non-defective
Actual	Defective	**80**	0
	Non-defective	5	**75**

Bold values represents the highest score

indicating the defective areas. However, there also exists a lot noise (white pixels) that do not correspond to the defect features. After the fifth convolution operation (i.e., Fig. 6f), the noise is eliminated, leaving the noticeable defect lines.

Table 6 further demonstrates that our proposed solution is an efficient method in classifying the type of leather defects. We summarize some of the remarkable previous works that are related to leather detection task, such as to: (a) differentiate between defective and non-defective images; (b) categorize the defect types, and; (c) recognize the leather type. There is one work that can exhibit perfect detection rate (i.e., # 5), but the sample size is just 16. However, it is difficult to draw a conclusion from the methodical approaches, as the experimental protocols, parameter settings and database used are different.

5.2 Localization performance using YOLOv2

Table 7 presents the segmentation performance for YOLOv2, with different mini-batch sizes (2, 4, 8, 16 and 32) and epochs values (30, 50, 100 and 150). The best average IoU achieved is 69.25%, when mini-batch size = 8 and epoch = 50. According to the segmentation results, it is able to recognize whether the sample has defect or not. Thus, it is can be considered as a type of detection. The performance metrics such as accuracy and F1-score are reported in Table 7 as well. The highest accuracy attained is 98.13%, when mini-batch size = 8 and epoch = 30. It is worth highlighting that the recall reaches 100% (all the defective leather images are correctly spotted), and its confusion matrix is provided in Table 8. The sample output using YOLOv2 are shown in Fig. 7a–c. It indicates that the defective regions have been successfully spotted with rectangular window boxes.

5.3 Localization performance using Faster R-CNN

Table 9 shows the localization performance using Faster R-CNN. Besides that, the best result obtained from YOLOv2 are compared. Note that the results shown in Table 9 are obtained by directly employing the localization task, without performing the detection stage in advance. We refer the experimental settings of (mini-batch size, epoch) = (8, 50) as YOLOv2-1 and (mini-batch size, epoch) = (8, 50) as YOLOv2-2. It can be seen that Faster R-CNN outperforms both of the YOLOv2 configurations. Faster R-CNN obtains an average IoU of 73.47%, which has + 4% compared to YOLOv2-1. The accuracy attained by Faster R-CNN is 96.88% , which is 1.25% lower than that of YOLOv2-2. It should also be noted that Faster R-CNN always operates at slower speed compared to YOLOv2. The confusion matrix for Faster R-CNN is shown in Table 10, which denotes an error rate of 3% (5 images with no defect are misjudged).

5.4 Training progress and computation time

Since there are three types of train/ test split configurations in the experiments, the duration required for the training and testing stages in each configuration is slightly different. Concretely, the exact time taken for the model training and testing is provided in Table 11. It is observed that the larger the training data, the lesser

Table 11 Time taken for the model training and testing for the classification task in different train/ test split configuration

Experiment	Training stage		Testing stage	
	# data sample	Time taken (s)	# data sample	Time taken (s)
Detection—[0.4 train/ test split]	224	10.44	336	0.57
Detection—[0.6 train/ test split]	336	17.63	224	0.46
Detection—[0.8 train/ test split]	448	24.75	112	0.34

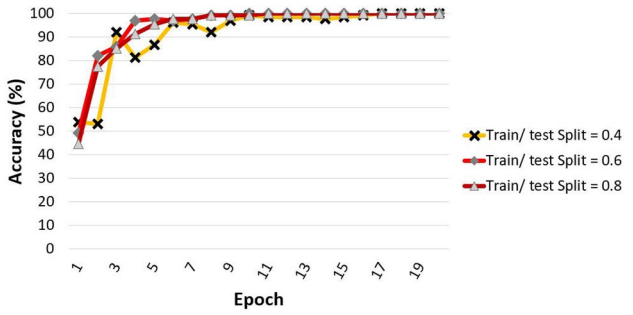

Fig. 8 The model training progress that portrays the classification accuracy of training data in each epoch

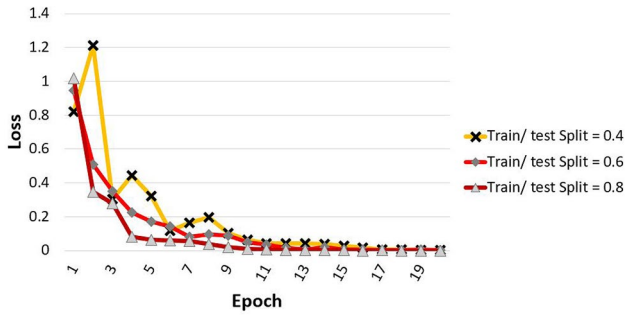

Fig. 9 The model training progress that portrays the classification loss of training data in each epoch

Table 12 Performance (%) of the complete system: detection (AlexNet + SVM) + localization

Method	Avg IoU	Acc	Rec	Pre	F1
Faster R-CNN	**73.47**	**100**	**100**	**100**	**100**
YOLOv2-1	69.24	**100**	**100**	**100**	**100**
YOLOv2-2	64.69	98.75	98.75	**100**	99.37

Bold values represents the highest score

the testing data. Thus, for the train/ test split of 0.8, it has the largest number of sample data (i.e., 448 images), the time taken for the model training is the longest (i.e., 25s). The training duration required reduces when the number of training data lesser, which requires 10s to complete the model training for train/test split of 0.4 that contains only 224 images. This phenomenon appears to be the same in the testing stage. Nonetheless, The training process is always time-consuming because the model repetitively adjusts the weights of the network until the training error is minimized. In contrast, the testing process performs a single run through the trained network. Thus, the time taken for the testing stage is significantly lesser (< 1s in all cases).

Table 5 provides the defect detection accuracy in three types of train/test splits (i.e., 0.4, 0.6, and 0.8). It can be seen that the detection accuracies yielded are similar (i.e., 97.02% ~ 98.66%). This implies that the trained network accommodates a relatively small data sample yet produces high detection accuracy. Figures 8 and 9 portray the sample training progress of the three scenarios in terms of accuracy and loss, respectively. Notice that, all cases illustrate that both the accuracy and loss metrics remain stagnant after the 10-th epoch.

5.5 Complete system performance

The overall performance for the proposed end-to-end leather defect detection and localization system is tabulated in Table 12.

The flowchart is shown in Fig. 2. AlexNet + SVM generates 100% score for the performance metrics of accuracy, recall, precision and F1-score. Then, to identify the defective regions, the average IoU obtained is 73.47%, resulting from the Faster R-CNN object detector.

In order to train a deep learning model that is optimally suited to our application, the Gradient-weighted Class Activation Mapping (Grad-CAM) (Selvaraju et al. 2017) visualization method is implemented in the network designed. It is used to examine the weights of the activations learned

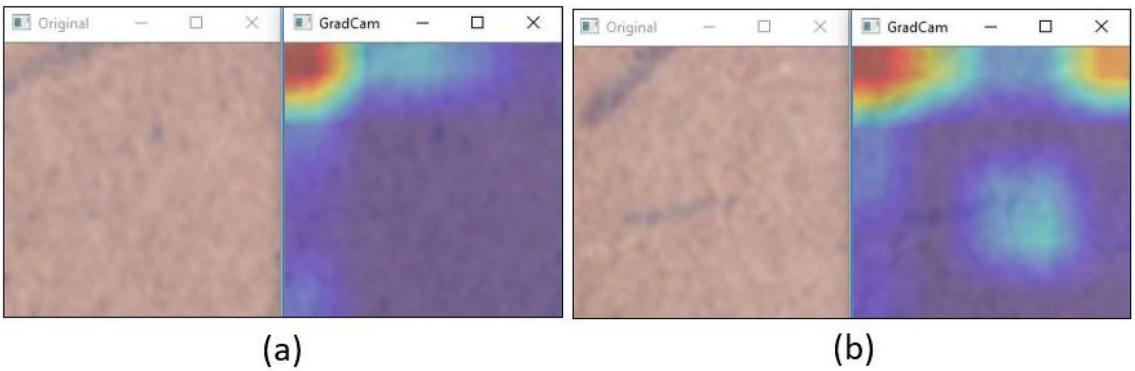

Fig. 10 Example of the original leather (left) and the output of Grad-CAM (right) for leather image that has **a** single defect and **b** two defects

throughout the network. The class-discriminative localization map Grad-CAM is the result of the linear combination of important weights from the feature maps A, and is denoted as:

$$L^{c}_{Grad-CAM} \in \mathbb{R}^{(u \times v)} = f(\Sigma_k \alpha^c_k A^k), \tag{18}$$

where u and v are the weight and height for each class c, respectively. The function f is the activation function for network. The α^c_k of class c at $k-th$ feature map is calculated by using the equation:

$$\alpha^c_k = \overbrace{\frac{1}{Z}\Sigma_i\Sigma_j}^{\text{global average pooling}} \underbrace{\frac{\delta y^c}{\delta A^K_{ij}}}_{\text{gradients via backpropagation}}, \tag{19}$$

where Z is a constant representing the number of pixels in the output feature map A, the output of class c (y^c) is defined as $\Sigma_k w^c_k \Sigma_i \Sigma_j A_{ij}$, such that w^c_k is the weights for class c at $k-th$ feature map and A_{ij} is the feature map at indexing position (i, j).

The output of the Grad-CAM enables us to provide important clues to fine-tune the parameters of the network that can effectively capture significant features. The example of the Grad-CAM images are shown in Fig. 10. The figure illustrates that the defects had been successfully detected using the Grad-CAM segmentation algorithm.

6 Conclusion

In conclusion, this paper suggests an approach to classify and localize the defective leather images. For the detection task, a pre-trained AlexNet as the feature extractor is applied, with SVM as the classifier. This method leads to a detection accuracy of 100%. In addition, the visualization of activation maps for each convolution layer in AlexNet shows that the features are well described. As for the localization stage, both the YOLOv2 and Faster R-CNN methods are tested. Faster R-CNN can achieve higher average IoU (73.47%) but has a slower computational speed compared to YOLOv2. Nonetheless, the network architecture training can be performed offline, and thus the defect identification is feasible to implement in a real-world situation.

In literature, this is a rather rare attempt in investigating the feasibility by adopting the deep learning models in the real leather samples provided by the industry. The leather defect detection performance in this study demonstrates the positive findings, albeit the evaluation of limited defect types. Besides, this study integrates the detection system such that the end-to-end mechanism consists of the procedures of image elicitation, image selection, image pre-processing, ground truth labeling, defect classification, and segmentation.

For future research, we may consider the algorithm improvement in three directions. First, pre-trained convolutional neural networks can be adopted to extract the defective area features, then utilize them for defect detection. Secondly, Single Shot Multibox Detector (SSD) segmentation tool can be used for defect detection, as it can provide accurate result with short train time. Last but not least, the issues with detection of multiple types of defects (i.e., insect bites, wrinkles, cuts, scratches, etc.) as well as on different leather types (cows, pigs, sheep, goats, and crocodiles) can be tackled.

Acknowledgements This work was funded by Ministry of Science and Technology (MOST), Taiwan (Grant No. MOST 109-2221-E-035-065-MY2, 109-2218-E-035 -002- and 108-2218-E-035-018-).

References

Aslam M, Khan TM, Naqvi SS, Holmes G, Naffa R (2020) Ensemble convolutional neural networks with knowledge transfer for leather defect classification in industrial settings. IEEE Access 8:198600–198614

Bong H-Q, Truong Q-B, Nguyen H-C, Nguyen M-T (2018) Vision-based inspection system for leather surface defect detection and classification. In: 2018 5th NAFOSTED Conference on Information and Computer Science (NICS), IEEE, pp 300–304

Deng J, Dong W, Socher R, Li L-J, Li K, Fei-Fei L (2009) Imagenet: a large-scale hierarchical image database. In: 2009 IEEE conference on computer vision and pattern recognition. IEEE, pp 248–255

Deng J, Liu J, Wu C, Zhong T, Gu G, Ling BW-K (2020) A novel framework for classifying leather surface defects based on a parameter optimized residual network. IEEE Access 8:192109–192118

Everingham M, Van Gool L, Williams CK, Winn J, Zisserman A (2010) The pascal visual object classes (voc) challenge. Int J Comput Vis 88(2):303–338

Girshick R (2015) Fast r-cnn. In: Proceedings of the IEEE international conference on computer vision, pp 1440–1448

He K, Zhang X, Ren S, Sun J (2016) Deep residual learning for image recognition. In: Proceedings of the IEEE conference on computer vision and pattern recognition, pp 770–778

Huang G, Liu Z, Van Der Maaten L, Weinberger K Q (2017) Densely connected convolutional networks. In: Proceedings of the IEEE conference on computer vision and pattern recognition, pp 4700–4708

Iandola F N, Han S, Moskewicz MW, Ashraf K, Dally WJ, Keutzer K (2016) Sequeezenet: Alexnet-level accuracy with 50x fewer parameters and< 0.5 mb model size. arXiv preprint arXiv:1602.07360

Jain AK, Duin RPW, Mao J (2000) Statistical pattern recognition: a review. IEEE Trans Pattern Anal Mach Intell 22(1):4–37

Jawahar M, Babu NC, Vani K (2014) Leather texture classification using wavelet feature extraction technique. In: 2014 IEEE International Conference on Computational Intelligence and Computing Research, IEEE, pp 1–4

Jawahar M, Babu NC, Vani K, Anbarasi LJ, Geetha S (2021) Vision based inspection system for leather surface defect detection using fast convergence particle swarm optimization ensemble classifier approach. Multimed Tools Appl 80(3):4203–4235

Jian L, Wei H, Bin H (2010) Research on inspection and classification of leather surface defects based on neural network and decision tree. In: 2010 International Conference On Computer Design and Applications, Vol 2, IEEE, pp V2–381

Kasi MK, Rao JB, Sahu VK (2014) Identification of leather defects using an autoadaptive edge detection image processing algorithm. In: 2014 International Conference on High Performance Computing and Applications (ICHPCA), IEEE, pp 1–4

Krizhevsky A, Sutskever I, Hinton GE (2012) Imagenet classification with deep convolutional neural networks. In: Advances in neural information processing systems, pp 1097–1105

Kwak C, Ventura JA, Tofang-Sazi K (2000) A neural network approach for defect identification and classification on leather fabric. J Intell Manuf 11(5):485–499

LeCun Y, Bottou L, Bengio Y, Haffner P (1998) Gradient-based learning applied to document recognition. Proc IEEE 86(11):2278–2324

Lin T-Y, Maire M, Belongie S, Hays J, Perona P, Ramanan D, Dollár P, Zitnick CL (2014) Microsoft coco: common objects in context. In: European conference on computer vision. Springer, pp 740–755

Liong S-T, Gan Y, Huang Y-C, Liu K-H, Yau W-C (2019a) Integrated neural network and machine vision approach for leather defect classification. arXiv preprint arXiv:1905.11731

Liong S-T, Gan Y, Huang Y-C, Yuan C-A, Chang H-C (2019b) Automatic defect segmentation on leather with deep learning. arXiv preprint arXiv:1903.12139

Mundt M, Majumder S, Murali S, Panetsos P, Ramesh V (2019) Meta-learning convolutional neural architectures for multi-target concrete defect classification with the concrete defect bridge image dataset. In: Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition, pp 11196–11205

Pistori H, Paraguassu WA, Martins PS, Conti MP, Pereira MA, Jacinto MA (2018) Defect detection in raw hide and wet blue leather. In: Computational Modelling of Objects Represented in Images. Fundamentals, Methods and Applications: Proceedings of the International Symposium CompIMAGE 2006 (Coimbra, Portugal, 20–21 October 2006). CRC Press, p 355

Ren S, He K, Girshick R, Sun J (2016) Faster R-CNN: towards real-time object detection with region proposal networks. IEEE Trans Pattern Anal Mach Intell 39(6):1137–1149

Ren S, He K, Girshick R, Sun J(2015) , Faster r-cnn: towards real-time object detection with region proposal networks. In: Advances in neural information processing systems, pp 91–99

Selvaraju RR, Cogswell M, Das A, Vedantam R, Parikh D, Batra D (2017) Grad-cam: visual explanations from deep networks via gradient-based localization. In: Proceedings of the IEEE International Conference on Computer Vision, pp 618–626

Serafim AL (1992) Object detection in images of natural scenes represented by ar models using laplacian pyramids: application to leather defects localization. In: Proceedings of the 1992 International Conference on Industrial Electronics, Control, Instrumentation, and Automation, IEEE, pp 716–720

Simonyan K, Zisserman A (2014) Very deep convolutional networks for large-scale image recognition. arXiv preprint arXiv:1409.1556

Statista (2019) Luxury leather goods. https://www.statista.com

Szegedy C, Liu W, Jia Y, Sermanet P, Reed S, Anguelov D, Erhan D, Vanhoucke V, Rabinovich A (2015) Going deeper with convolutions. In: Proceedings of the IEEE conference on computer vision and pattern recognition, pp 1–9

Tschandl P, Rosendahl C, Kittler H (2018) The ham10000 dataset, a large collection of multi-source dermatoscopic images of common pigmented skin lesions. Sci Data 5(1):1–9

Ventura JA, Chen J-M (1996) A structural model for shape recognition using neural nets. J Intell Manuf 7(1):1–11

Villar P, Mora M, Gonzalez P (2011) A new approach for wet blue leather defect segmentation. In: Iberoamerican Congress on Pattern Recognition. Springer, pp 591–598

Winiarti S, Prahara A, Murinto DPI (2018) Pre-trained convolutional neural network for classification of tanning leather image. Network (CNN) 9(1)

Zhang X, Zhou X, Lin M, Sun J (2018) Shufflenet: an extremely efficient convolutional neural network for mobile devices. In: Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition, pp 6848–6856