

Research Article

Automated Classification System for Tick-Bite Defect on Leather

Y. S. Gan,¹ Wei-Chuen Yau,² Sze-Teng Liong ,³ and Chih-Cheng Chen ^{4,5}

¹School of Architecture, Feng Chia University, Taichung 407, Taiwan

²School of Computing and Data Science, Xiamen University Malaysia, Jalan Sunsuria, Bandar Sunsuria, 43900 Sepang, Selangor Darul Ehsan, Malaysia

³Department of Electronic Engineering, Feng Chia University, Taichung 40724, Taiwan

⁴Department of Automatic Control Engineering, Feng Chia University, Taichung 40724, Taiwan

⁵Department of Aeronautical Engineering, Chaoyang University of Technology, Taichung 413, Taiwan

Correspondence should be addressed to Sze-Teng Liong; stliong@fcu.edu.tw and Chih-Cheng Chen; ccc@gm.cyut.edu.tw

Received 4 February 2021; Revised 5 April 2021; Accepted 25 August 2021; Published 16 February 2022

Academic Editor: Teen-Hang Meen

Copyright © 2022 Y. S. Gan et al. This is an open access article distributed under the Creative Commons Attribution License, which permits unrestricted use, distribution, and reproduction in any medium, provided the original work is properly cited.

Natural leather is a durable, breathable, stretchable, and pliable material that comes in various styles, colors, finishes, and prices. It is an ideal raw material to manufacture luxury products such as shoes, dresses, and luggage. The leather will be categorized into different grades that are determined by visual appearance, softness, and natural defects. This grading process requires a manual visual inspection from experienced experts to ensure proper quality assurance and quality control. To facilitate the inspection process, this paper introduces an efficient automated defect classification framework that is capable to evaluate if the sample patches contain defective segments. A six-step preprocessing procedure is introduced to enhance the quality of the leather image in terms of visibility and to preserve important features representation. Then, multiple classifiers are utilized to differentiate between defective and nondefective leather patches. The proposed framework is capable to generate a classification accuracy rate of 94% from a collection of samples of 1600 pieces of calf leather patches.

1. Introduction

Leather is the most popular raw material made from animal skins (i.e., cow, lamb, deer, elk, pig, etc.) Amongst them, most leather is made from cowhide as it is relatively easier to acquire in large quantities and its thick characteristics make it desirable for various types of products. Nevertheless, each leather piece comes with some imperfections that may result in the grain surface or structure of a hide. Common unsightly appearance existing on natural leather surfaces include scars, flay cuts, vainness, and irregular coloring. The surface appearance of a leather piece is an important indicator to determine its grading and hence affecting the selling price. To date, the conventional method to manually inspect the quality of the leather pieces is still adopted in the industrial manufacturing process.

In brief, the basic procedures to convert the raw animal hides into leather are as follows: (1) Soaking: to remove the dirt and curing salts by immersing the leather in water for several hours to several days; (2) liming: to remove the

epidermis, hair, and subcutaneous materials; (3) tanning: to create the protein cross-links in the collagen by penetrating the chemicals into the hides; (4) drying: to get rid of excess water; and (5) dyeing: to produce the desired custom color. Some of the natural defects are inconspicuous before the tanning process and they gradually appear to be apparent during the leather finishing process. On the other hand, those defective areas with minor damage will be repaired and roughed up with fillers to create a smooth and even surface. Finally, the finishing leather pieces will be graded before shipping to the customer.

The grading process is one of the most critical and exhausting procedures because it involves a manual assessment to visually inspect the defective parts of the leather. Particularly, the type of defect (i.e., cuts, wrinkles, and scabies), the defective size, and the severity level (i.e., critical, major, minor, or trivial) are the major aspects of quality control. The inspectors require to conduct a thorough manual evaluation on the same piece of leather multiple

times, viewing from multiple angles, distances, and lighting conditions to ensure the correctness and completeness. However, it should be noted that each judgment is subjective as it highly relies on the individual. Thus, human inspection is costly, time-consuming, inefficient, and inconsistent. It can be prone to human mistakes or errors due to this boring and repetitive task, or when the labor is feeling stressed and rushing to complete the task. Therefore, there is vital to design an automatic leather defect inspection system in order to improve the grading and inspection processes, in the meantime cutting off unnecessary costs.

The final goal of this paper is to classify the leather sample into either defective and nondefective classes. In brief, the four primary contributions of this research work are summarized as follows:

- (1) Proposal of six preprocessing steps and XBoost ANN classifier to categorize the defective leather patch
- (2) Verification the robustness of the proposed algorithm by validating them on several distinct machine learning classifiers
- (3) Comprehensive experimental evaluation and comparative analysis are carried out on over 1600 leather images
- (4) Demonstration of the promising classification results by reporting both the qualitative and quantitative findings

The subsequent sections of the paper are arranged as follows. A review of related literature is presented in Section 2. Then, Section 3 describes the proposed framework in detail which includes the intuition and explanation of the principal image processing techniques exploited. The experimental design such as the details of the database used, performance metrics, and the configuration of the parameters in the experiment are presented in Section 4. The classification performance is presented and discussed in Section 5 with further analysis. Finally, the conclusion is drawn in Section 6, accompanied by methodological recommendations for future research.

2. Literature Review

To date, the literature that carried out the automatic classification or segmentation tasks on the leather pieces is yet limited [1–3]. Besides, the experimental data are varied and hence it is difficult to make a fair test of performance to verify the effectiveness of the proposed methods. For instance, reference [4] collects the leather patch dataset using a robot arm such that each image is captured under consistent lighting source, same viewing angle, and distance. In total, the dataset contains 584 images. Then, a series of procedures are introduced to localize the tick-bite defects on leather patches. Succinctly, a segmentation algorithm, namely, Mask Region-based Convolutional Neural Network (Mask R-CNN) is adopted to learn the local features from 84 defective images. As a result, a classification accuracy of $\sim 70\%$ is obtained when evaluated on 500 testing images.

Later, reference [5] employs the same data elicitation process to collect a different piece of calf leather. In brief, 27

images are collected and each piece is partitioned into 24 small patches. Thus, in total, 648 images are used in the experiment. Different from reference [4, 5] conducts both the classification and segmentation processes to predict two types of defects, namely, black lines and wrinkle. A transfer learning technique is adopted to fine-tune the parameters in AlexNet architecture for the classification task, whereas UNet architecture is employed for the segmentation task. As a result, the classification performance attained is 95% and the segmentation task obtained an Intersection over Union rate of close to 100%. However, it should be noted that the black line and wrinkle defects are relatively obvious and occupy a larger region. Thus, a reasonably higher classification result can be achieved.

Reference [6] designs a statistical approach based on the image intensity to tackle the classification task for both the datasets released by references [4, 5]. Briefly, this work adopts simple statistical features operations such as mean, variance, skewness, kurtosis, lower, and upper quartile values. Then a feature selection method of the 2-sample Kolmogorov–Smirnov test is exploited to determine meaningful features. Then, three methods are applied to eliminate redundant features: percentile thresholding, Gaussian mixture model (GMM), and K-means clustering. Finally, seven types of classifiers are adopted to differentiate between the defective and nondefective leather patches. The best classification accuracy generated are 99% and 77% on two different datasets (i.e., [4, 5]), respectively. In short, this paper successfully outperforms reference [4] by 7% while obtaining a comparable performance with reference [5].

On the other hand, conventional methods such as feature extraction and reduction are adopted for leather defect detection task in which deep learning methods are not applied. For example, reference [7] utilizes the FisherFace feature reduction technique to project the local features of the leather images from high-dimensional image space to a lower-dimensional feature space in order to effectively distinguish the targeted classes. Concisely, the feature size of each image sample has been reduced from 4202 to 160. The extracted features include the attributes of color details, histograms of the color, co-occurrence matrix, Gabor filters, and the original pixels. To validate the effectiveness of the proposed method, the experiment was tested on 2000 samples that are composed of seven defective classes. Then, three types of classifiers are employed to predict the defective type. The best classification accuracies obtained are $\sim 88\%$ for wet blue and $\sim 92\%$ for rawhide images.

On the other hand, a leather type classification task was performed by reference [8] that evaluated 1000 leather sample images to differentiate among monitor lizard, crocodile, sheep, goat, and cow. Despite each leather type may contain samples with different colors, the proposed method is capable to distinguish the texture and characteristics of each leather type. Thus, a 99.9% classification accuracy was achieved by adopting the pretrained AlexNet architecture. However, no defect inspection or defect classification task is involved in the experiment.

Based on the aforementioned discussion, the research works conducted thus far are manageably finite. Inspired by

reference [4, 6], this paper aims to enhance the classification performance by introducing a simple yet effective solution. Particularly, the type of defect class in this classification task is strictly limited to only the tick bite. In brief, six preprocessing steps are applied to improve the images and to extract the local information of the leather patches. Next, the feature sets are fed into several two-class classifier models independently by exposing the relationships between the encoded features, in order to generate corresponding predicted labels. The classifiers involved in the experiment herein include decision tree, SVM, k -NN, Artificial Neural Network (ANN), XBoost ANN, and others which are employed to categorize the testing data.

3. Proposed Method

There are two major steps proposed in the algorithm, namely, the preprocessing and classification. The flowchart of the process is portrayed in Figure 1. Concisely, the images are first passed to a series of preprocessing steps, such as histogram matching, resizing, grayscale, Canny edge detection, Gaussian blurring, and histogram of the gradient. On the other hand, the classification task employs state-of-the-art supervised classifiers such as decision tree [9], discriminant analysis [10], SVM [11], k -Nearest Neighbor (NN) [12], Artificial Neural Network (ANN), XBoost Artificial Neural Network, and others.

The details of mathematical derivations of the aforementioned preprocessing methods and classifiers are elaborated in Section 3.1 and Section 3.2, respectively.

3.1. Preprocessing Procedure. The six preprocessing techniques employed in the experiment are shown in Figure 2 and each step is described as follows. In addition, sample images are shown in Figure 3 to illustrate the effect in each preprocessing step.

Step 1. Histogram Matching

The images have performed a histogram matching with the ground truth template image to standardize the new image so that to eliminate the difference of brightness or contrast due to the environmental situation. The idea is to map the probability density function $P_r(r)$ of the original image into the desired output $P_z(z)$, where r and z are intensity values of color spaces such as HSV/HLS, YUV, and YCbCr. The mapping is built by finding the best matches $P_z(z)$ for each input in $P_r(r)$ such that satisfied the following equation:

$$|P_r(r) - P_z(z)| = \min_k |P_r(r) - P_z(k)|. \quad (1)$$

Step 2. Resizing

The image is then resized from 400×400 to 100×100 . This downsampling step is to minimize the computational complexity; meanwhile, the execution speed is increased. Besides, it reduces the background noise.

Step 3. Gray Scale

The image is converted from color into a grayscale image. It can minimize redundancy and dimensionality; thus the computational requirements are also reduced.

Step 4. Gaussian Filter

A Gaussian blur is applied to smooth the background area and the defective area. Gaussian blur transforms each pixel in the image to produce normally distributed pixel values by its local neighbor through a mathematical function defined as follows:

$$G(x, y) = \frac{1}{2\pi\sigma^2} e^{-x^2+y^2/2\sigma^2}, \quad (2)$$

where x is the distance from the center to the horizontal axis, y is the distance from the center to the vertical axis, and σ is the standard deviation of the Gaussian distribution. The values from this distribution are used to build a convolution matrix that is applied to the original image. Furthermore, by using a suitable filter size, it will produce more vulnerability intensity of new pictures. If the areas in an image are the same, it will generate a similar intensity. Hence, it increases the discriminant effect of the defective and nondefective areas.

Step 5. Canny Edge Detection:

Up to this step, the defective and nondefective areas should be easier to differentiate. The image is then enhanced by focusing on the gradient difference in the intensity of images. Succinctly, the process of the Canny edge detection algorithm can be implemented using these five steps:

- (i) A Gaussian filter is adopted to remove the image noise and suppress the meaningless information.
- (ii) The intensity gradients of the image are obtained by applying the edge detector operators like Sobel, Prewitt, and Robert.
- (iii) A nonmaximum suppression is employed to eliminate the spurious response such as spikes or noises.
- (iv) The lower and upper threshold values are specified to identify potential edges.
- (v) The edges are tracked by hysteresis whereby the *weak* edges that are not connected to *strong* edges are minimized.

The experiment conducted in this paper considers the Sobel operator. It applies convolution on the image with a separable, integer, and small-valued filter in the horizontal/vertical directions. Particularly, the Sobel operator approximates the gradient of the image by applying convolution on the image with a separable kernel in either horizontal or vertical directions. In general, 3×3 kernels are used on both the horizontal and vertical derivative approximation that are denoted as G_x and G_y , respectively:

$$G_x = \begin{bmatrix} -1 & 0 & 1 \\ -2 & 0 & 2 \\ -1 & 0 & 1 \end{bmatrix}, G_y = \begin{bmatrix} 1 & 2 & 1 \\ 0 & 0 & 0 \\ -1 & -2 & -1 \end{bmatrix}. \quad (3)$$

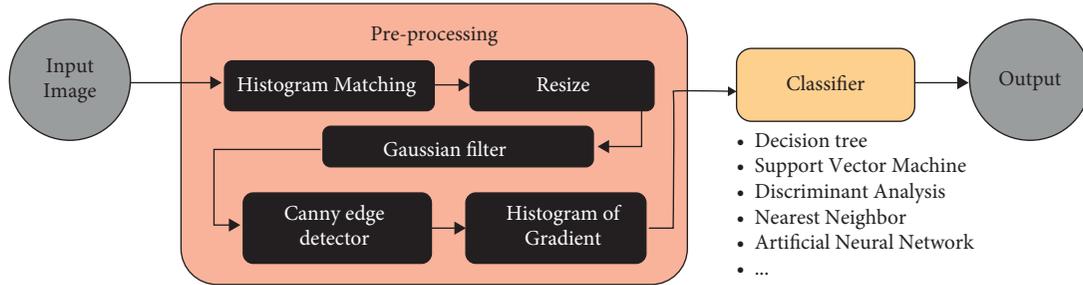


FIGURE 1: The proposed tick-bite defect classification system.

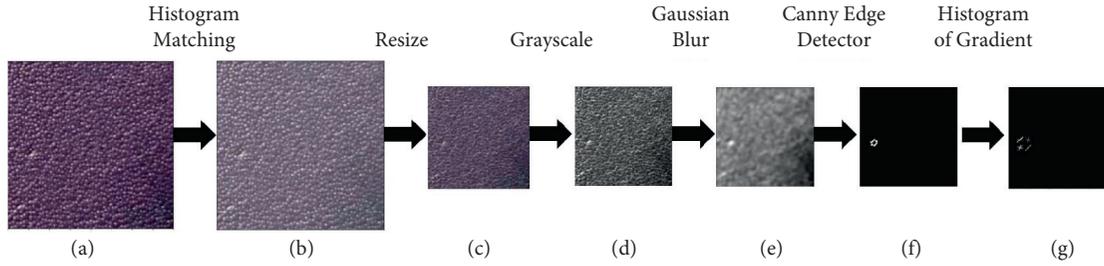


FIGURE 2: The example of after applying the steps of preprocessing methods: (a) original image; (b) histogram matching; (c) resizing; (d) grayscale; (e) Gaussian blur; (f) Canny edge detection; and (g) histogram of gradient.

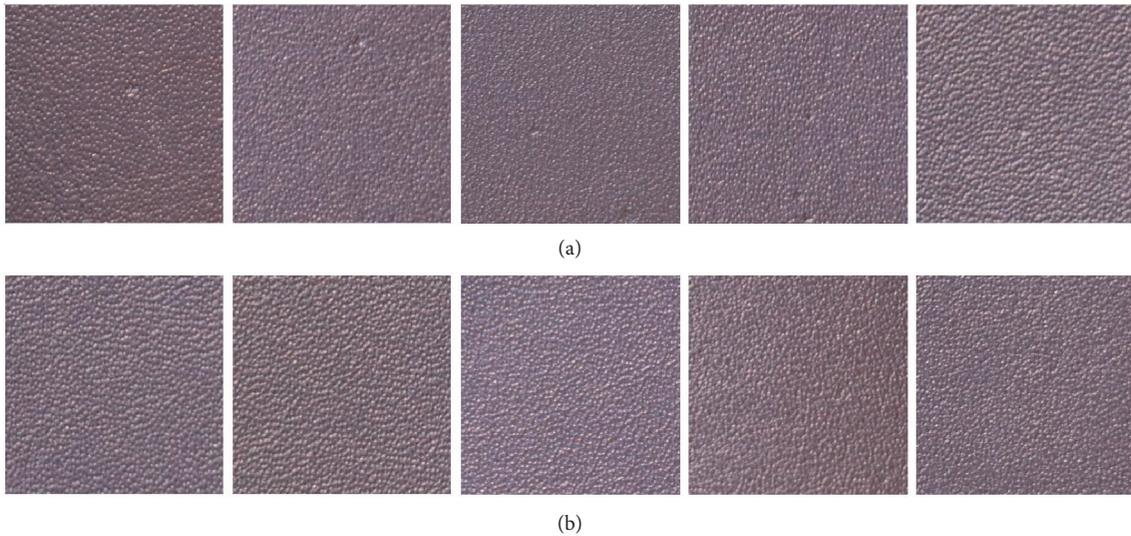


FIGURE 3: Sample leather images that contain (a) defect and (b) no defect.

At each regularly spaced sample points, the gradient approximations can be derived and represented to the gradient's magnitude (ρ) and gradient's direction (θ):

$$\rho = \sqrt{G_x^2 + G_y^2}. \quad (4)$$

$$\theta = \tan^{-1}\left(\frac{G_y}{G_x}\right). \quad (5)$$

Step 6. Histogram of Gradient

The Canny edge detector in Step 5 returns a binary image. It is then split into 8×8 pixels per cell to calculate the histogram of gradient of the binary images. This reveals the frequency histogram gradient of the orientation of the edges at each local patch, especially for the defective areas. On a uniform grid of cells, HOG summarizes the intensity gradients based on their respective directions to derive the local appearance features that describe the focus information of the corresponding image. Then, the histogram of gradient directions within the connected cells is concatenated such that an enriched resultant feature vector is constructed. Owing to the advantages of the HOG descriptor such as fast

computation speed and effectiveness in encoding the local shape information, it is one of the feature extractors that had been widely adopted in the research community. Specifically, the steps to realize the HOG algorithm are as follows.

3.1.1. Gradient Image Generation. The image filtering method can be applied by utilizing the kernels that contain both the horizontal and vertical kernels, namely, $[-1\ 0\ 1]$ and $[-1\ 0\ 1]^T$ sliding window. Concisely, the kernel convolutes with the original image from left-right, to top-bottom. Then, (4) and (5) is applied to acquire the pixel-wise magnitude and orientation maps. As a result, the regions with constant and similar color intensity are eliminated from the image; in the meantime, the important outlines or edges are kept without change.

3.1.2. HOG Computation in $m \times n$ Cells. Each image is partitioned into $m \times n$ cells such that a more compact feature representation can be constructed. Thus, a 9-bin histogram that falls in the angles range of $0, 20, 40, 60, \dots, 180$ is computed.

3.1.3. Cell's Blocks Normalization. The magnitude computed may be vulnerable and sensitive to changes in illumination. Therefore, a simple normalization operation is implemented locally for each block. Finally, the resultant feature vector is enriched by concatenating all the histograms.

3.2. Classifier. This subsection briefly elaborates on the characteristics of classifiers. Concretely, both the functions of the conventional classifiers (i.e., decision tree, SVM, NN, discriminant analysis, etc.) and the ANN are stated in Subsections 3.2.1 and 3.2.2.

3.2.1. Conventional Classifier. After attaining the feature vectors from the feature descriptors discussed in the previous section, they are then processed by the classifiers to distinguish the defective status. Some widely known classifiers that available in Sklearn Package are utilized, namely, decision tree, SVM, NN, and ensemble classifier. Note that the classifiers adopted herein are supervised machine learning approach:

- (1) *k*-Nearest Neighbor (k-NN) [12]: this is one of the simplest classifiers as it is easy to implement and no training time is required. The predicted outcome is identified based on the simple majority vote system and determination of the number of nearest neighbors.
- (2) Support Vector Classification (SVC) [13]: it can be used for either the classification or regression analysis. It involves at least a quadratically fitting scale with the number of samples.
- (3) Linear Support Vector Machine (SVM) [11]: a linear kernel is utilized to project the input data to a higher

dimensional space. This data transformation process finds an optimal boundary between the possible outcomes.

- (4) Decision tree [9]: it builds a classification model by adopting simple decision rules. A tree-structured model is created by outlining all the possible consequences. In brief, the decision tree consists of the root, nodes, branches, and leaves. The predicted response is generated by following the decision from the root node down to the leaf node.
- (5) Random forest [14]: it is a collection of simple tree estimator that process various subsamples of the dataset and obtain the average values to boost the classification accuracy and prevent over-fitting.
- (6) Multilayer perceptron (MLP) [15]: it composes three basic layers, namely, the input, hidden, and output layers. Each layer may contain a different number of the neuron. Specifically, the neurons in the input layer depend on the dimension of the input data. The number of neurons in the hidden layer is subjective as it relies on the function's complexity and the attribute properties of the targeted classes. Finally, the number of neurons in the output layer is the number of output classes.
- (7) Adaptive Boosting (AdaBoost) [16]: it is a meta-estimator that learns a single "strong classifier" from several "weak" classifiers. It produces a set of optimal features that consider the weights factor before the combination of the classifiers.
- (8) Discriminant analysis [10]: it has a quadratic decision boundary to develop discriminant functions to examine the difference between the predictor variables.
- (9) Extreme gradient boosting [17]: it trains many weak prediction models sequentially and ensembles them. The typical models are decision trees and the learning procedure generalizes the new model to provide a more accurate and optimized predictor.

3.2.2. Artificial Neural Network (ANN) Learning Features. ANN is a significant part of artificial intelligence as it mimics the computational principles of neural networks of an animal. Owing to its remarkable generalization capability and promising correlation-based feature selector, it has been extensively used in the research field such as handwritten text recognition [18], weather forecasting [19], financial economics [20], and agricultural land assessment [21].

Basically, ANN incorporates three layers, namely, the input, hidden, and output layers. Concisely, the neurons in both the hidden and output layers adopt the sigmoid activation functions in performing the backpropagation operation. The output of the ANN can be acquired by the following equation:

$$y_{\text{output}} = \frac{1}{1 + \exp^{-(b_2 + W_2 (\max(0, b_1 + W_1 * X_n)))}}, \quad (6)$$

where W_1, W_2, b_1, b_2 are weights and biases parameters and X_n refers to the data input. The Adam optimization algorithm [22] is adopted to adaptively update the learning rates during the model training. In addition, we propose the “XBoosting ANN” by implementing an extreme gradient boosting method onto the ANN outputs.

4. Experiment

4.1. Database. The experimental data adopt the database released by reference [4]. Concretely, the database consists of 1605 leather patches that have the size of $90 \times 60 \text{ mm}^2$ (width \times length). Amongst them, 503 images contain one or more tick-bite defects, whereas the remaining 1102 images are nondefective images. In brief, all the images are collected using a 6-axis articulated robot arm DRV70L from Delta, which load-bearing capacity is 5 kg. The robot arm is equipped with a Canon 77D camera fitted with a 135 mm focal length lens. Each captured data has an image resolution of $2400 \times 1600 \text{ pixels}^2$. A lightning source is utilized to guarantee a consistent brightness distribution on the leather pieces. A screenshot of the experimental setup is illustrated in Figures 3 and 4 which shows the samples for the defective and nondefective images.

An illustrative example that describes the bounding box with the estimated size is shown in Figure 5. Besides, the largest and the smallest defect samples are depicted in Figure 6.

4.2. Experiment Configuration. In the classification stage, 5-fold cross-validation is applied to test the unseen data. The dataset is by the first split to a ratio of 7 : 3 for the train: test subsets. Then, the training subset is further partitioned into 7 : 3 into train and validation subsets. Therefore, the final division of the dataset is about a ratio of 5 : 3:2 for the train: test:validation subsets, which consists of 785 : 483 : 337 images, respectively. Concretely, the train features will be fed into the classification model; in the meantime, the validation images are utilized in order to determine the optimal experimental configuration and parameter settings (i.e., filter size of the Gaussian filter and threshold value for the Canny edge detector). Finally, the refined model is used to validate the test images, and the performance metrics are described in the following subsection.

4.3. Performance Metrics. This is a binary classification problem where the output should produce the label of “defect” or “no defect.” Thus, the following four metrics can be derived from the 2×2 confusion matrix:

$$\text{Accuracy} := \frac{TP + TN}{TP + FP + TN + FN}, \quad (7)$$

where TP is the predicted pixel that has correctly identified the defective pixel; TN is the nondefective pixel that has been correctly predicted; FP is the pixel that is incorrectly predicted as defective pixel; and FN is the undetected defective pixels.



FIGURE 4: Illustration of the experimental setup.

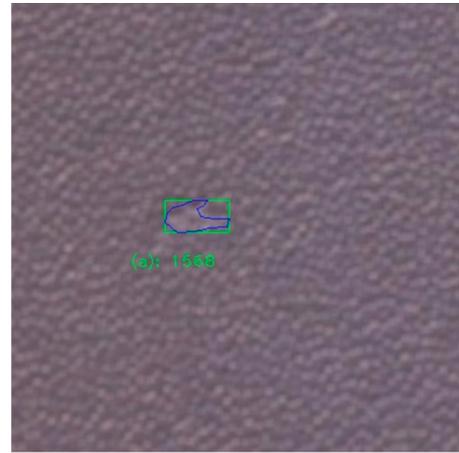


FIGURE 5: The surface area of the defect with a bounding box.

On the other hand, F1-score performance metric is computed:

$$F1 - \text{score} := 2 \times \frac{\text{Precision} \times \text{Recall}}{\text{Precision} + \text{Recall}}, \quad (8)$$

for

$$\text{recall} := \frac{TP}{TP + FN}, \quad (9)$$

and

$$\text{precision} := \frac{TP}{TP + FP}. \quad (10)$$

There are two types of F1-score, namely, macro-averaged and weighted-average. The former is simply the mean of per class F1-score, which is similar to the macro-averaged precision and macro-averaged recall, which are calculated by the mean of precision and recall, respectively. For the weighted-average F1-score, the weighted-precision and weighted-recall are calculated by considering the weights to each class:

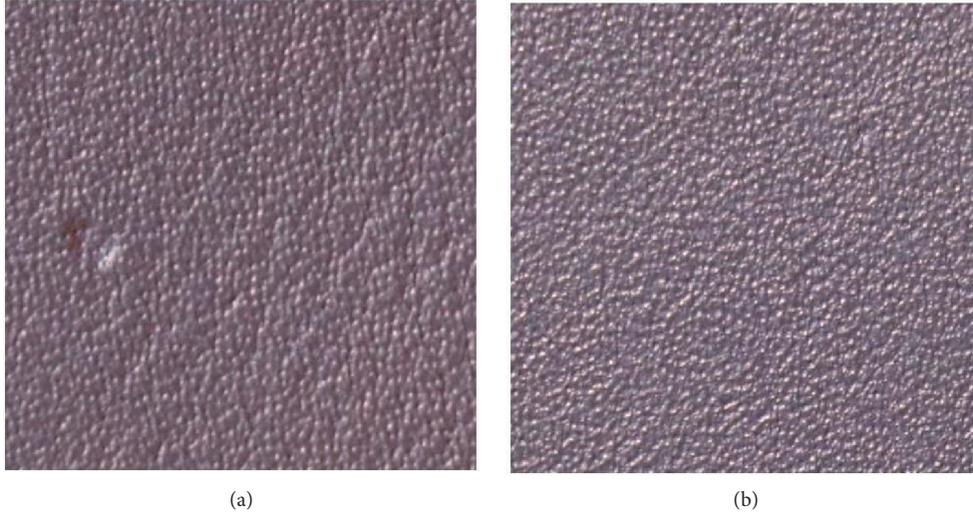


FIGURE 6: The defective samples that has the (a) largest and (b) smallest tick-bite defects.

$$\text{weighted - average } F1 - \text{score} = \frac{(w_1 S_1 + w_2 S_2 + \dots + w_n S_n)}{(w_1 + w_2 + \dots + w_n)}, \quad (11)$$

where w_i is the number of samples for class i and S_i belongs to F1-score, precision, and recall for each class i .

5. Result and Discussion

It should be noted that for the Canny edge detector, different size of Gaussian filters will detect differently sized features in the input image and producing distinct-sized feature maps. To seek an optimal configuration of the edge detector, its kernel size is first fixed to the common sizes, namely, (5,5), (7,7), and (9,9), respectively. Then, the range of the threshold values is set to [80, 230]. The validation accuracy of the classification accuracy against the threshold parameter in the Canny edge detector is portrayed in Figure 7. There is a similar trend for the kernel sizes of 7 and 9, whereby the best results are obtained when the threshold is minimal (i.e., 84 to 104). In contrast, when kernel size = 5, a low threshold did not outperform, compared to that of the kernel of 7 and 9. In addition, it can also be observed that when kernel size = 5, the highest accuracy (when threshold = 160) obtained is relatively lower.

From the preliminary result performed in Figure 7, we opt to select the optimal threshold of 92 for all the kernel sizes throughout the remaining of the experiments. The results of the accuracy when adopting ANN and XBoost ANN are shown tabulated in Table 1, with the detailed TP, FP, FN, TN, and F1-score. It can be seen that when the preprocessing steps do not involve HoG (the first three rows), the accuracy in the ANN classifier is 69%, while higher accuracy is attained when utilizing XBoost ANN (up to 82%). On the contrary, when HoG is added as one of the preprocessing steps, all the accuracies in both the ANN and XBoost ANN improved. Specifically, a promising classification result of 94% is exhibited when kernel size = 7.

To further analyze the impact of the HoG in the preprocessing step, a receiver operating characteristic (ROC) can illustrate the effectiveness of the statistical model of the classifier. Particularly, when HoG is not applied as one of the preprocessing steps, the ROC curve is shown in Figure 8. It is observed that the accuracy of the micro-average ROC is 69%, whereas the macro-average ROC is 50%. On the other hand, when HoG is included in the proposed method, the accuracies of the ROC improved up to 95%, as demonstrated in Figure 9.

In addition, we opt for ANN and Xboost ANN as the classifiers in our experiment. The reason being is because other classifiers are not outperforming based on the features extracted, especially in this binary classification task. The classification results are summarized in Figure 10. It can be seen that both the ANN and Xboost ANN achieve an accuracy of more than 90%. For SVM with linear kernel and random forest classifiers, their results are promising as well ($\sim 90\%$). Other classifiers like k-NN, SVC, decision tree, AdaBoost, gradient boosting, and discriminant analysis seem not suitable to be adopted in this experiment.

The proposed framework is compared to three other works that performed the binary classification on the same leather dataset. Concretely, the three other methods utilized the ANN [4], AlexNet [4], and statistical analysis [6] as the key feature descriptors. The results comparison is summarized in Table 2 whereby the metrics presented are accuracy and F1-score. It is interesting to highlight that the deep learning network such as AlexNet does not perform well in this classification task. This may in part due to the overfitting phenomena. With a relatively small dataset and highly imbalanced data, where the number of the nondefective images is doubled of the defective ones, the network is not able to generalize well and thus leads to a poor classification result. Notably, the results generated in this paper outperform the state-of-the-art, in which the accuracy and F1-score reported are both 94%. However, it should be noted that the training images considered in the experiment herein are almost half of

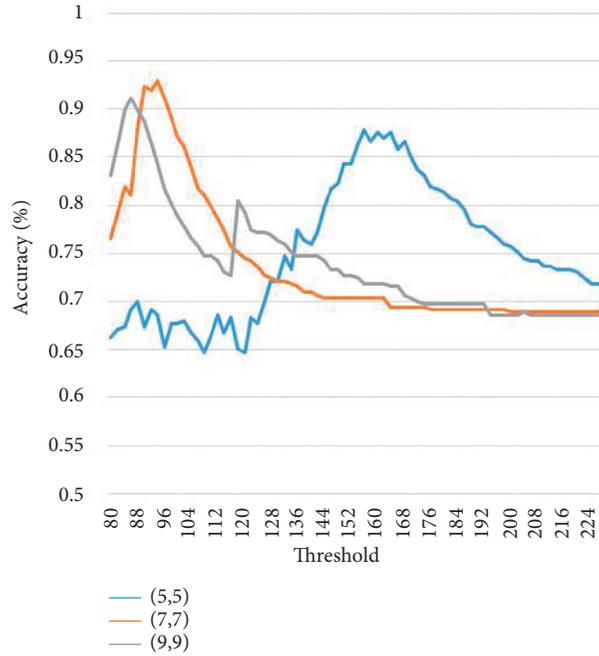


FIGURE 7: The validation accuracy of the classification accuracy against the threshold parameter in the Canny edge detector.

TABLE 1: Comparison of results when employing ANN (A) and XBoost ANN (X) as the classifiers.

Methods	TP		FP		FN		TN		Accuracy		Macro-avg		Weighted-avg	
	A	X	A	X	A	X	A	X	A	X	F1-score		F1-score	
											A	X	A	X
<i>w/o</i> HoG (5,5)	331	266	0	65	151	114	0	37	0.69	0.63	0.41	0.52	0.56	0.61
<i>w/o</i> HoG (7,7)	331	274	0	57	151	48	0	103	0.69	0.78	0.41	0.75	0.56	0.78
<i>w/o</i> HoG (9,9)	331	281	0	50	151	35	0	116	0.69	0.82	0.41	0.8	0.56	0.83
With HoG (5,5)	314	313	17	18	50	49	101	102	0.86	0.86	0.83	0.83	0.86	0.86
With HoG (7,7)	325	325	6	6	21	25	130	126	0.94	0.94	0.93	0.92	0.94	0.93
With HoG (9,9)	323	323	8	8	41	41	110	110	0.90	0.90	0.87	0.87	0.89	0.89

The bold numbers represent the highest values within the experimental results presented.

the dataset. Nonetheless, with the utilization of neural network architecture and gradient features, it achieves unprecedented improvements in the classification results.

In a nutshell, this paper proposes a new feature enhancement pipeline in classifying the defective leather image. Specifically, a large portion of the contribution is attributed to the preprocessing stage, in which the processes include histogram matching, resizing, grayscale normalization, Gaussian blurring, and Canny edge detection. Thereafter, the defective region becomes clearer and

noticeable. The HOG descriptor is utilized to convert the image into a 1D feature vector. Finally, multiple classifiers are employed to evaluate the robustness of the proposed mechanism. As a whole, the proposed method requires relatively lower computational resources whilst achieving promising classification accuracy of up to 94%. A brief comparison of the state-of-the-arts is provided in Table 3 to show the primary difference in tackling this leather defective classification problem. Note that the total number of leather samples for the experiments is slightly different.

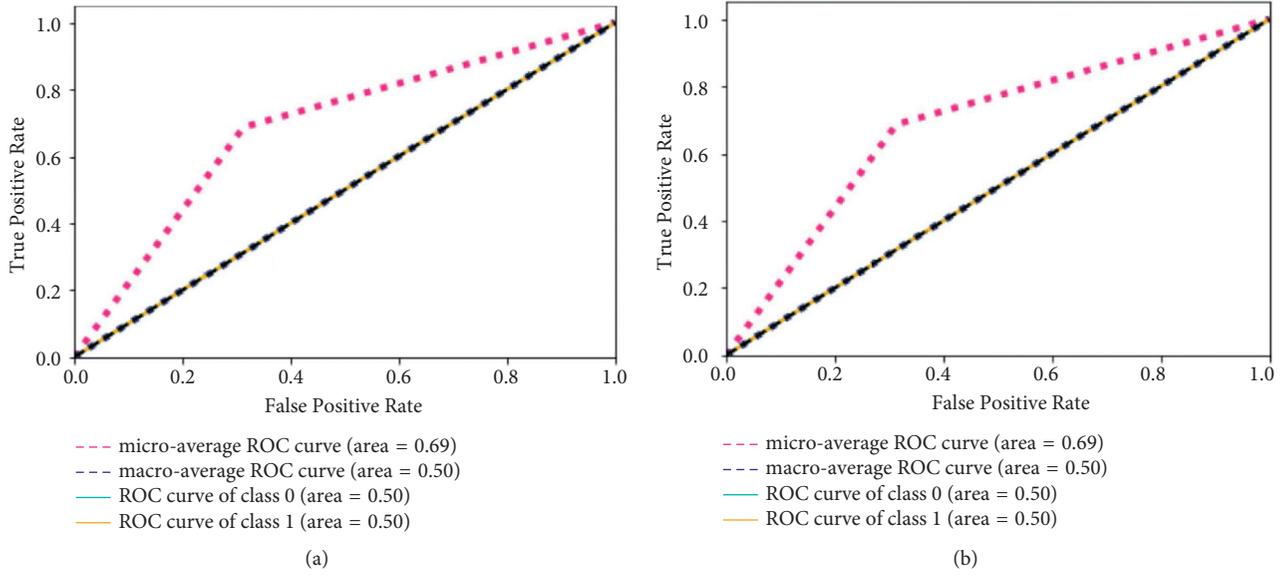


FIGURE 8: ROC curve when histogram of gradient is not included as one of the preprocessing steps, when evaluated on (a) validation set and (b) testing set.

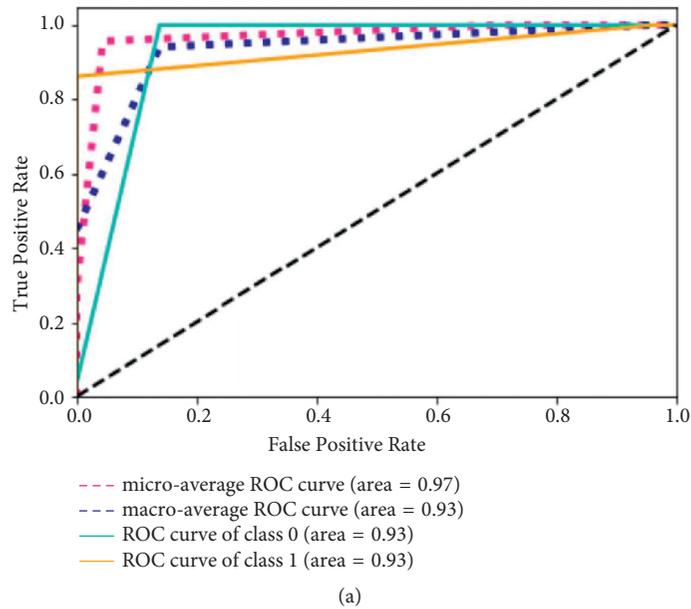


FIGURE 9: Continued.

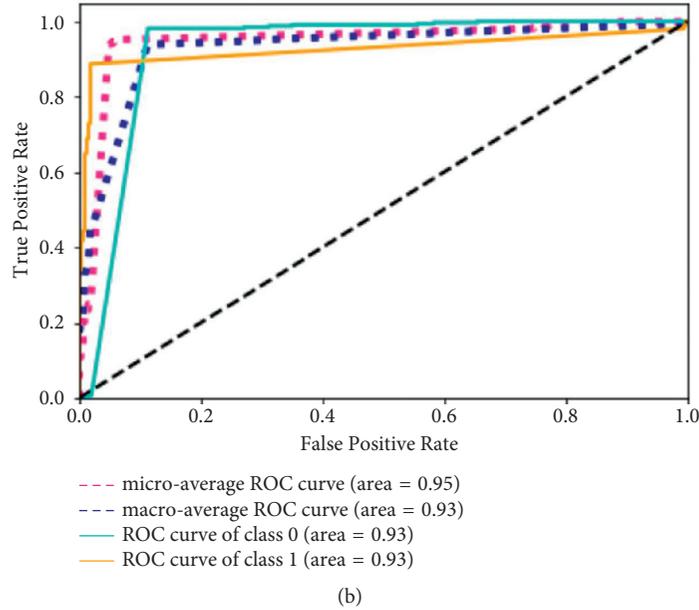


FIGURE 9: ROC curve when applying histogram of gradient as one of the preprocessing steps, when evaluated on (a) validation set and (b) testing set.

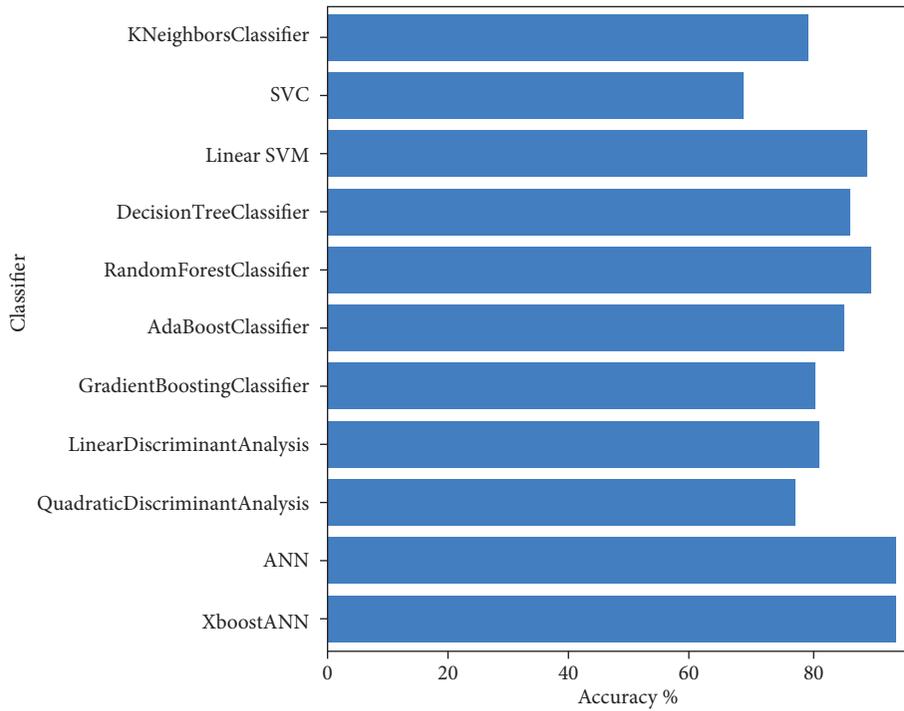


FIGURE 10: Classification performance of the proposed method when employing different classifiers.

TABLE 2: Classification performance comparison to the state-of-the-arts on the same dataset.

Method	Defective: nondefective	Train: validation: testing split	Method	Accuracy (%)	F1-score (%)
[4]	370 : 1527	60 : 5:35	ANN	80	89
[4]	233 : 699	90 : 0:10	AlexNet	74	84
[6]	503 : 1102	70 : 0: 30	Statistical analysis + SVM	74	97
Ours	503 : 1102	49 : 21: 30	XBoost ANN + HOG	94	94

The bold numbers represent the highest values within the experimental results presented.

TABLE 3: Classification performance comparison to the state-of-the-arts on the same dataset.

Method	Preprocessing	Feature Extractor	Classifier	Accuracy (%)
[6]	Grayscale normalization, Gaussian filter	Statistical features (mean, variance, skewness, kurtosis, lower, and upper quartile value)	SVM	74
[23]	Grayscale normalization, resizing, Canny edge detection, block partition		ANN	80
[23]	Grayscale normalization, resizing, Canny edge detection, block partition		AlexNet	76
[4]	—		Mask R-CNN	80
Ours	Histogram matching, resizing, grayscale normalization, Gaussian blurring, and Canny edge detection	Histogram of gradient	k-NN, SVC, SVM, MLP, AdaBoost, decision tree, random forest, discriminant analysis, extreme gradient boosting	94

The bold numbers represent the highest values within the experimental results presented.

6. Conclusion

This study introduced a binary classification system to distinguish if a leather image contain tick-bite defect. Thorough experiments and analyses have been conducted to verify the robustness of the proposed algorithm. Overall, promising results are exhibited when exploiting a series of preprocessing methods and two neural network classifiers. As a result, the best classification accuracy obtained is 94% when employing ANN and XBoost ANN as the classifiers. As this experiment strictly limited to the defect type of tick-bite defect, potential direction for future research in this area includes the development of classification or segmentation system to determine the defect types such as open cuts, closed cuts, wrinkles, holes, and scabies. Apart from investigating the defect type, the experiment can be extended to evaluate on other leather type such as the hides of other animals lamb, crocodile, and snakes. Ultimately, a fully automated hardware setup that consists of the functions of capturing leather image patches, identifying the defective areas, and of laser cutting of the leather can be developed in the future.

Data Availability

The nature of the data in an Excel File, and the data and the code can be accessed on the following website: <https://github.com/christy1206/XBoost-leather>. There are no restrictions on data access. The [Excel File] data used to support the findings of this study are included in the article.

Conflicts of Interest

The authors declare that they have no conflicts of interest.

Acknowledgments

This work was funded by Ministry of Science and Technology (MOST) (Grant Number: MOST 109-2221-E-035-065-MY2, MOST 110-2221-E-035-052).

References

- [1] M. Aslam, T. M. Khan, S. S. Naqvi, G. Holmes, and R. Naffa, "On the application of automated machine vision for leather defect inspection and grading: a survey," *IEEE Access*, vol. 7, Article ID 176065, 2019.
- [2] M. Jawahar, N. K. Babu, K. L. J. A. Vani, L. J. Anbarasi, and S. Geetha, "Vision Based Inspection System for Leather Surface Defect Detection Using Fast Convergence Particle Swarm Optimization Ensemble Classifier Approach," *Multimedia Tools and Applications*, vol. 80, no. 3, pp. 4203–4235, 2020.
- [3] M. K. Kasi, J. B. Rao, and V. K. Sahu, "Identification of leather defects using an autoadaptive edge detection image processing algorithm," in *Proceedings of the 2014 International Conference on High Performance Computing and Applications (ICHPCA)*, pp. 1–4, IEEE, Bhubaneswar, India, December 2014.
- [4] S. T. Liong, Y. S. Gan, Y. C. Huang, C. A. Yuan, and H. C. Chang, "Automatic Defect Segmentation on Leather with Deep Learning," March 2019, <https://arxiv.org/abs/1903.12139>.
- [5] S.-T. Liong, D. Zheng, Y.-C. Huang, and Y. Gan, "Leather defect classification and segmentation using deep learning architecture," *International Journal of Computer Integrated Manufacturing*, vol. 33, no. 10-11, pp. 1105–1117, 2020.
- [6] Y. Gan, S.-S. Chee, Y.-C. Huang, S.-T. Liong, and W.-C. Yau, "Automated leather defect inspection using statistical approach on image intensity," *Journal of Ambient Intelligence and Humanized Computing*, vol. 12, pp. 1–17, 2020.
- [7] W. P. Amorim, H. Pistori, M. C. Pereira, and M. A. C. Jacinto, "Attributes reduction applied to leather defects classification," in *Proceedings of the 2010 23rd SIBGRAPI Conference on Graphics, Patterns and Images*, pp. 353–359, IEEE, Gramado, Brazil, August 2010.
- [8] S. Winiarti, A. Prahara, and D. P. I. Murinto, "Pre-trained convolutional neural network for classification of tanning leather image," *Network (CNN)*, vol. 9, no. 1, 2018.
- [9] S. R. Safavian and D. Landgrebe, "A survey of decision tree classifier methodology," *IEEE transactions on systems, man, and cybernetics*, vol. 21, no. 3, pp. 660–674, 1991.
- [10] W. R. Klecka, G. R. Iversen, and W. R. Klecka, *Discriminant analysis*, Sage, California, CA, USA, 1980.

- [11] J. A. K. Suykens and J. Vandewalle, "Least squares support vector machine classifiers," *Neural Processing Letters*, vol. 9, no. 3, pp. 293–300, 1999.
- [12] T. Cover and P. Hart, "Nearest neighbor pattern classification," *IEEE Transactions on Information Theory*, vol. 13, no. 1, pp. 21–27, 1967.
- [13] S. R. Gunn, "Support vector machines for classification and regression," *ISIS technical report*, vol. 14, no. 1, pp. 5–16, 1998.
- [14] A. Liaw and M. Wiener, "Classification and regression by randomforest," *R News*, vol. 2, no. 3, pp. 18–22, 2002.
- [15] M. W. Gardner and S. Dorling, "Artificial neural networks (the multilayer perceptron)—a review of applications in the atmospheric sciences," *Atmospheric Environment*, vol. 32, no. 14–15, pp. 2627–2636, 1998.
- [16] G. Rätsch, T. Onoda, and K.-R. Müller, "Soft margins for adaboost," *Machine Learning*, vol. 42, no. 3, pp. 287–320, 2001.
- [17] T. Chen, T. He, M. Benesty et al., *Xgboost: Extreme Gradient Boosting, R Package Version*, vol. 1, no. 4, pp. 1–4, 2015.
- [18] S. España-Boquera, M. J. Castro-Bleda, J. Gorbe-Moya, and F. Zamora-Martinez, "Improving offline handwritten text recognition with hybrid hmm/ann models," *IEEE Transactions on Pattern Analysis and Machine Intelligence*, vol. 33, no. 4, pp. 767–779, 2011.
- [19] J. W. Taylor and R. Buizza, "Neural network load forecasting with weather ensemble predictions," *IEEE Transactions on Power Systems*, vol. 17, no. 3, pp. 626–632, 2002.
- [20] Y. Li and W. Ma, "Applications of artificial neural networks in financial economics: a survey," vol. 1, pp. 211–214, in *Proceedings of the 2010 International symposium on computational intelligence and design*, vol. 1, IEEE, Hangzhou, China, October 2010.
- [21] F. Wang, "The use of artificial neural networks in a geographical information system for agricultural land-suitability assessment," *Environment and Planning A: Economy and Space*, vol. 26, no. 2, pp. 265–284, 1994.
- [22] D. P. Kingma and J. Ba, "Adam: A Method for Stochastic Optimization," December 2014, <https://arxiv.org/abs/1412.6980>.
- [23] S.-T. Liong, Y. S. Gan, K.-H. Liu et al., "Efficient neural network approaches for leather defect classification," June 2019, <https://arxiv.org/abs/1906.06446>.