Journal of Manufacturing Systems 51 (2019) 52–60

Contents lists available at ScienceDirect

Journal of Manufacturing Systems

journal homepage: www.elsevier.com/locate/jmansys

Machine vision intelligence for product defect inspection based on deep learning and Hough transform

Jinjiang Wang[a,*], Peilun Fu[a], Robert X. Gao[b]

[a] School of Mechanical and Transportation Engineering, China University of Petroleum, Beijing, 102249, China
[b] Department of Mechanical and Aerospace Engineering, Case Western Reserve University, Cleveland, OH, 44106, USA

ARTICLE INFO

Keywords:
Defective product inspection
Machine vision
Deep learning
Hough transform
Inverted residual block

ABSTRACT

Machine vision based product inspection methods have been widely investigated to improve product quality and reduce labour costs. Recent advancement in deep learning provides advanced analytics tools with high inspection accuracy and robustness. However, the construction of deep learning model is typically computationally expensive, which may not match the requirements for quick inspection. Therefore, this paper presents a new deep learning based machine vision inspection method to identify and classify defective product without loss of accuracy. In specific, Gaussian filter is first performed on the acquired image to limit random noise. Then, a region of interest (ROI) extracting project is conducted based on Hough transform to remove the unrelated background, thereby offloading the computational burden of the subsequent identification process. The construction of the identification module is based on convolutional neural network, whereas inverted residual block is introduced as the basic block to strike a good balance between identification accuracy and computational efficiency. The superior inspection performance is obtained using the proposed method with a large amount of dataset which consists of defective and defect-free bottle images.

1. Introduction

Smart manufacturing refers to a new manufacturing paradigm where manufacturing machines are fully connected by network, monitored by sensors, and controlled by advanced computational intelligence to improve product quality, system productivity, and sustainability [1,2]. As a powerful tool of product quality controlling, machine vision based system has been widely investigated and applied, which can range from object measurement [3,4], target identification [5], vision based control [6,7] and defective product inspection [8,9]. In terms of defective product inspection, based on the result and confidence of inspection, the controller of the inspection system will decide whether to re-inspect, repair or discard the inspected product, thereby preventing low quality products reach the market and cutting down labor investments. Generally, an inspection system is composed of the following modules: image acquisition, image processing, feature extraction and decision making [10]. This paper emphasizes on the construction of the latter three modules, which is considered as the core of an inspection system [11].

Features and shallow machine learning based inspection method has been widely adopted for analyzing acquired image. With the help of advanced image processing algorithm, properties of product defects, like color, shape and texture, can be extracted from images. An expert system or intelligent classifier can be built in further for classification. One primary challenge with such a method is how to obtain concise and defect-related vector representation of the captured image, namely feature extraction. The feature quality quite depends on the experience and ability of system designer; improper extraction may lead to a high miss-detection rate or false alarm rate. Moreover, the designed feature extractor is subject to the application scenario. Once the manufacturing process changes, the previously designed feature extractor may no longer be suitable.

Deep learning has received substantial interest in recent years, which uses a cascade of multiple layers of nonlinear processing units for feature extraction, and each layer regards the output of the previous layer as input [12]. With such structure, it is possible to integrate feature extraction and classification into one framework and optimizes them jointly, where the final layer is used to output expected label and the others are used for feature extraction. Benefiting from its enormous model scale, elaborate network structure and effective optimization algorithm, the initialized deep neural network is able to learn how to directly and hierarchically draw generic features from raw captured images in a supervised or unsupervised manner, thus achieving promising product inspection performance and generality. Vision system

* Corresponding author.
E-mail address: jwang@cup.edu.cn (J. Wang).

https://doi.org/10.1016/j.jmsy.2019.03.002
Received 7 December 2018; Received in revised form 13 March 2019; Accepted 29 March 2019
Available online 12 April 2019

designers could focus on collecting defective product samples rather than developing indicators for each type of defect.

In the most cases, an inspection system should be able to quickly respond to the product in the inspection area. However, in practical application, deep learning based methods may not be the best solution for image analysis due to their high computing power requirements. Regardless the overfitting problem, the accuracy of a deep learning model generally increases with the growth of model size, which will unavoidably slow down the processing speed. The situation is more complicated when multimodal vision information (e.g. thermal image, depth image and hyperspectral image, etc.) is provided to empower neural network to learn a complex recognition task in a complementary manner. Since each modality typically needs an individual network branch, the computational burden is multiplied inevitably. In summary, it is not easy to make a good trade-off between inspection accuracy and computational efficiency when putting a deep learning model into practice.

To address the aforementioned problems, this paper introduces a deep learning based approach to realize fast product inspection with guaranteed accuracy. The proposed efficient approach consists of three main phases. In order to keep the consistency of the captured images and get rid of the influences of noise, Gaussian filtering is first performed. The second phase removes the unrelated background content based on probabilistic Hough transform to avoid unnecessary calculations in the final image identification stage. Based on the received region of interest (ROI), feature extraction and decision making are combined through a lightweight deep neural network in the final phase, where an inverted residual block is introduced as the foundation. Previous works have demonstrated some promising solutions to maximize the capabilities of hardware devices, like graphics processing units (GPU) based parallel computing [13] and fog computing [14,15]. While this paper focuses on developing an efficient image processing and identification pipeline for an online inspection system, which is not in conflict with such parallel processing technologies.

Our specific case study is related to the production of bottled wine. As finished bottles move along the conveyor belt successively, the CCD camera captures images for each bottle and uploads them to the server for processing. An illustration of the objects to be inspected is presented in Fig. 1. All the bottles must be cleaned and completed before being packed and any imperfect or contaminated bottle should be inspected to advise on quality management, or directly instructed to the control unit.

The rest of this paper is organized as follows. Section II reviews the related works on machine vision based industrial inspection methods and systems. Section III elucidates the developed inspection method. Experiments and inspection results are presented in Section IV. Also, comparative and empirically analyses are discussed in this section. Finally, Section V gives the conclusion.

2. Related work

Feature based methods have been used for many years in the field of industrial inspection and recognition. In [16], 71 features are extracted from 3D and color 2D data to indicate the quality of slate slab, e.g. surface uniformity, material defects and warping. Zeng et al. proposes a man-made clear visual feature based on strobe light to highlight the edges between the seam and metal on weld; then the accurate seam edges can be computed by thresholding and edge extraction [18]. Sparse coding is utilized in [17] to obtain non-background features from original visual features generated by scale-invariant feature transform, histogram of oriented gradients (HOG), etc. In [8], Retinex algorithm based illumination correction method is used to enhance the features extracted from the COMS image of the micro-multilayer aspherical lens. In [9], a new approach for threshold selection is proposed to find the optimal threshold value for segmentation by comparing the histogram modes of the background and defective regions, then the sensitive features can be extracted based on produced binary image. Based on the extracted features, a shallow machine learning model or an inference system could be built in further to obtain the inspection result.

As an efficient and interpretable method, template based detection also has played an important role in industrial vision system [19]. By matching and comparing defect-free template image and test image, the defect can be revealed clearly. In [20], the expectation–maximization algorithm is employed to find mutual edge points in both compared images by assigning weights to individual edge points, thus achieving a precision alignment between template and test image. Bag-of-words (BOW) is used for planar products template selection in [21], and the approximate maximal clique algorithm is then employed for image registration. A contour-based image registration method and an improved fuzzy c-means cluster based segmentation method are proposed in [22] to detect defective mobile phone screen glass.

Instead of manually designing a task-related inspection pipeline, deep neural network builds it toward self-learning from image samples. Among various deep neural networks, convolutional neural networks (CNN) is particularly conspicuous as it can naturally model the spatial relationship in an image by applying a large amount of filters. In the following, some previous works on CNN based inspection methods are reviewed. In [23], a multitask CNN is proposed to integrate wire defect region detection and defective type classification. A deep-structured learning model is developed in [24], which employs randomized general regression network and CNN to detect single-defect and mixed-defect patterns respectively. A LED cup aperture detection system is developed in [25] based on three-point fitting and CNN regression network. The need for a large labeled dataset may limit its application. Fortunately, as more and more factories focus on the accumulation of defective product data, this issue can be mitigated. Moreover, many viable solutions have been found in literatures, including data generation and augmentation [26], transfer learning [27], unsupervised learning [28] and semi-supervised learning [29].

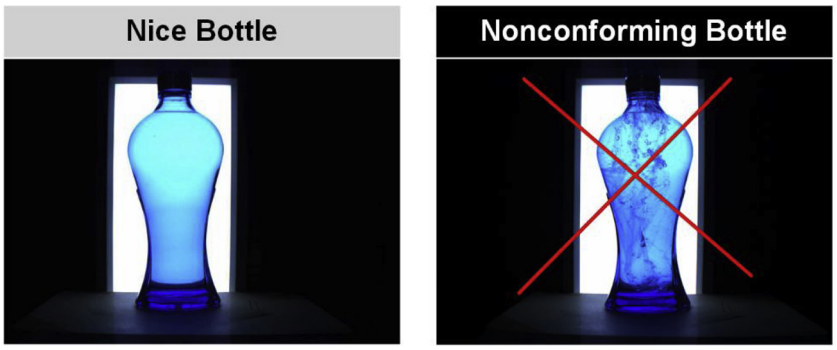

Fig. 1. Defective and defect-free product samples in our case.

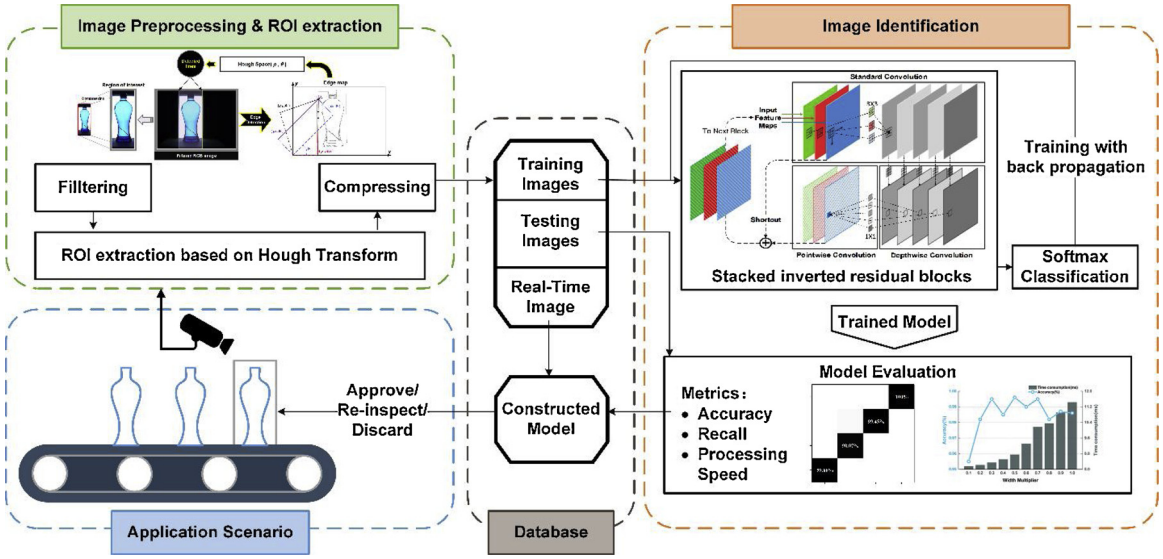

Fig. 2. The framework of the proposed method.

There may be problems if the local images are directly fed to a neural network, mainly unnecessary computation brought by background content and noise interference. In addition, training and applying a satisfactory neural network are typically computational expensive. The inspection method presented in this paper is able to cover both accuracy and efficacy by combining conventional image processing methods and a lightweight deep neural network.

3. Methodology

In this paper, an efficient product inspection method is proposed to strike a balance between recognition accuracy and overall model complexity. The completed image processing and identification pipeline is summarized in Fig. 2. As the conveyer belt moved on, product images can be captured and uploaded to the server for further analysis. With the proposed method, defective products can be detected through three major stages: image preprocessing, ROI extraction and image identification. The image preprocessing stage aims at getting a better representation of the raw image through Gaussian filter and canny edge detector. Then Hough Transform maps the edge graph to the Hough space to find the boundary of the light source. The detected boundary can be used as the basis for extracting ROI of the filtered image. The compressed ROI is stored in the database while being sent out of the image identification module.

The image identification module is constructed based on inverted residual block, which is a good alternative to standard convolutional block, reducing model size and computation time while maintaining identification accuracy. Since the proposed inspection method is a data driven approach, the system designer should first focus on collecting labeled defect and defect-free product sample images, and organizing them into dataset. The image dataset is randomly divided into two parts: training set and testing set, for training and evaluation respectively. The training phase continually corrects the network parameters in a supervised manner to make the prediction errors as small as possible. However, bad hyperparameter settings, like excessive network width, depth and iterations number, may lead to an overfitting problem where images other than training images cannot be accurately identified. Accordingly, model evaluation is of significant to search a good network structure and training strategy [30], thereby obtaining a practical identification model and storing it. In the application phase, the deserialized model is able to analyze processed real time image and give feedback on current production status. Specific illustrations about the three stages are presented in the following sections.

3.1. Image preprocessing

Online inspection system typically relies on the continuous running of imaging device, which inevitably leads to an increasing temperature in imaging device. Excessive imaging device temperature may cause the captured images to be corrupted by Gaussian noise, thereby decreasing the confidence and accuracy of the inspection. Therefore, the captured images are firstly blurred by a Gaussian function, which is a commonly used method to limit the influence of noise and enhance the performance of the subsequent image recognition module. With a 3×3 Gaussian filter, the output pixel value $C_{i,j}$ can be calculated as follows:

$$C_{i,j} = \frac{1}{9} \sum_{m=-1}^{1} \sum_{n=-1}^{1} W_{i+m,j+n} Z_{i+m,j+n} \tag{1}$$

$$W_{i+m,j+n} = \frac{1}{2\pi\sigma^2} e^{-\frac{(i+m)^2+(j+n)^2}{2\sigma^2}} \tag{2}$$

where $Z_{i+m,j+n}$ denotes to the input pixel value. W denotes to the weight at the corresponding position of the filter, which is produced by a Gaussian function. σ is a pre-set standard deviation. Empirically, σ is controlled by the size of Gaussian. The Gaussian kernel independently scans image on different color channels to ensure that all pixels are processed.

3.2. Region of interest extraction

The captured images do not guarantee that all contained information is task-related as the irrelevant background content may occupy a considerable part of the image, resulting in unnecessary calculations in the next identification stage. In practice, there is typically a lighting source to outstand defected products and ensure the consistency of the captured images. Fortunately, such a lighting source is usually in the shape of rectangle or round. For this reason, probabilistic Hough transform is employed to quickly detect the line or circle of the lighting source edge; then the detection result could be utilized for extracting ROI from filtered image.

Noted that a rectangular or circular area with a fixed size and anchor point may also be a potential solution to the problem of ROI extraction. However, the foremost problems of such a method is the fact that the tiny variations of camera angle, imaging instruction, and takt time cannot be well handled. The position of the product to be inspected may shift due to such factors, further causing the ROI extraction procedure to fail.

In this paper, Hough transform is performed to detect the edge lines of a lighting source, where the filtered RGB image is first converted to grayscale. The grayscale image is then processed by the canny edge detector as an edge map in further. Canny detector [31] first calculates the edge gradient G and gradient direction θ for each pixel as follows:

$$G = \sqrt{G_x^2 + G_y^2}$$
$$\theta = \arctan(G_x/G_y) \tag{3}$$

where G_x and G_y denote to the edge gradient calculated by an edge operator, such as Sobel and Prewitt, in the horizontal and vertical direction, respectively. Canny detector uses non-maximum suppression, threshold filtering and weak edge tracking to clear the primary edge map. Noted that the edge map is a binary map, where a pixel with a value of one denotes to an edge point.

Hough transform [32] is then performed to map edge points to Hough space (ρ,θ), which is defined as follows:

$$\rho = x\cos\theta + y\sin\theta,\ \theta \in (0, \pi) \tag{4}$$

where x and y denote to the coordinate of the edge point. The (ρ,θ) pair jointly describes a straight line passing through the edge point (x,y). ρ is the distance from the origin to the line, and θ is the angle, as illustrated in Fig. 3. By quantizing the Hough space and accumulating the (ρ,θ) pairs produced by all the edge points, the accumulation matrix can be obtained. The first few (ρ,θ) pairs with the largest cumulative number are selected and transformed inversely to the image space as the detected lines, which can be written as:

$$y = \frac{-\cos(\theta)}{-\sin(\theta)}x + \frac{\rho}{\sin(\theta)} \tag{5}$$

To accelerate the inspection process, a fast version of Hough Transform named Progressive Probabilistic Hough Transform [33] is employed to detect lines in practice. In this work, the detected lines are used to extract ROI from raw RGB images to remain product appearance information as the contamination of a bottle is also reflected in the color space. However, the utilization of grayscale images or edge maps may be more efficient in some application scenarios due to their smaller volumes. It depends on the type of product to be inspected. Finally, the extracted ROI is compressed and fed into the image identification module. The framework of image preprocessing and ROI extraction is shown in Fig. 3.

3.3. Image identification

The aim of this stage is to recognize the detailed class of the extracted ROI. In order to achieve high classification accuracy and speed, the image identification module is constructed by cascading inverted residual blocks which are proposed in [34]. The well-structured RGB image sample is first fed into a standard convolutional layer to extract primary features and expand the channels by applying different convolutional filters to each input channel, which can be expressed as:

$$\mathbf{F}^{(m)} = \sum_{n=1}^{N} \mathbf{W}^{(n,m)} * \mathbf{C}^{(n)} + b^{(n)} \tag{6}$$

where $\mathbf{F}^{(m)}$ denotes to the m^{th} output feature maps, N denotes to the number of input channels. $\mathbf{W}^{(n,m)}$ is a convolutional filter corresponding to the n^{th} input channel and m^{th} output channel of which the weights are optimized during backpropagation. Empirically, each optimized filter is believed to capture one of the visual features in the input image, such as a corner, and an edge in an orientation or other patterns. * denotes to the convolutional operation. Instead of employing a pooling layer to get the compressed representation of a feature map, we implicitly set the step of convolutional operation to two to lessen calculation burden of the subsample process.

Then, a set of cascaded inverted residual blocks are built for further visual feature extraction. The basic structure is shown in Fig. 4: a standard convolutional layer followed by a depthwise separable convolutional layer, and the final output of the block is a summation of the input feature maps and the output feature maps. Depthwise separable convolutions [36] accelerate feature extraction and save model scale with acceptable prediction accuracy decreasing by decomposing standard convolutional operation into two steps: depthwise convolution and pointwise convolution.

The first standard convolutional layer receives input feature maps from previous block or convolutional layers and expands channels. For the generated feature maps in different channels, depthwise convolution merely applies one convolutional filter to each feature map to capture visual feature. Accordingly, the number of the output channels keeps the same with the input. Pointwise convolution is a special case of standard convolution, whose kernel size is set to one to linearly combine different input channels. Beyond that, pointwise convolution also compacts the channels of the input to match the dimension to the input of the inverted residual block, and the feature maps to which the input is element-wise added will be the final output of the block. Such connections across several layers are called shortcut, which can well handle the gradient vanishing problem in a deep neural network [37]. The information flow in an inverted residual block is formulated as follows, where the biases are omitted for brevity.

$$\mathbf{X}^{(n,l)} = \mathbf{X}^{(n,l-1)} + \sum_{m=1}^{M} \mathbf{W}_{1\times1}^{(m)} * \left[\mathbf{W}_{3\times3}^{(m)} * \sum_{n=1}^{\alpha N} (\mathbf{W}_{3\times3}^{(n,m)} * \mathbf{X}^{(n,l-1)}) \right] \tag{7}$$

where $\mathbf{X}^{(n,l-1)}$ and $\mathbf{X}^{(n,l)}$ denote to the n^{th} generated feature maps in the l-1^{th} and l^{th} blocks, respectively. $M = t\alpha N$, where t is the expansion factor and α is the width multiplier introduced to conveniently control the size of the model. Almost all the convolutional layers are followed by a batch normalization [35] and a Relu6 nonlinearity. However, the

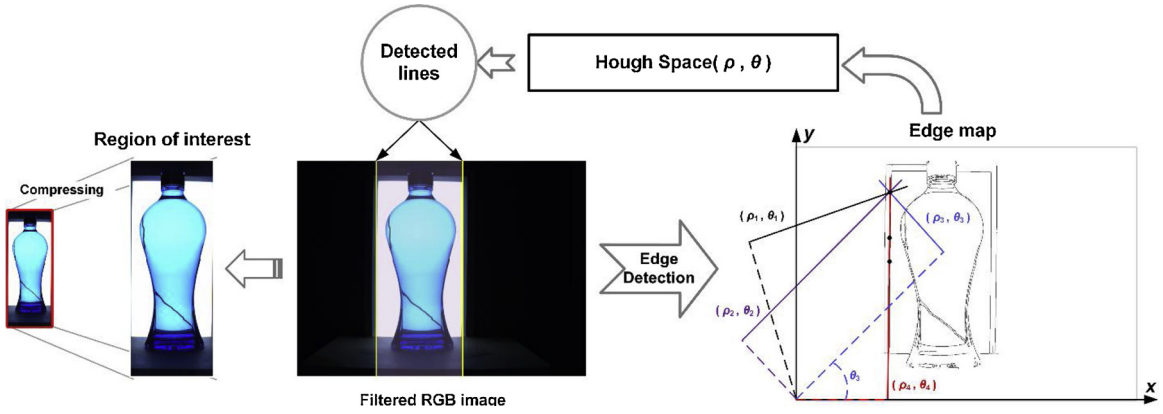

Fig. 3. Steps of image preprocessing and ROI extraction.

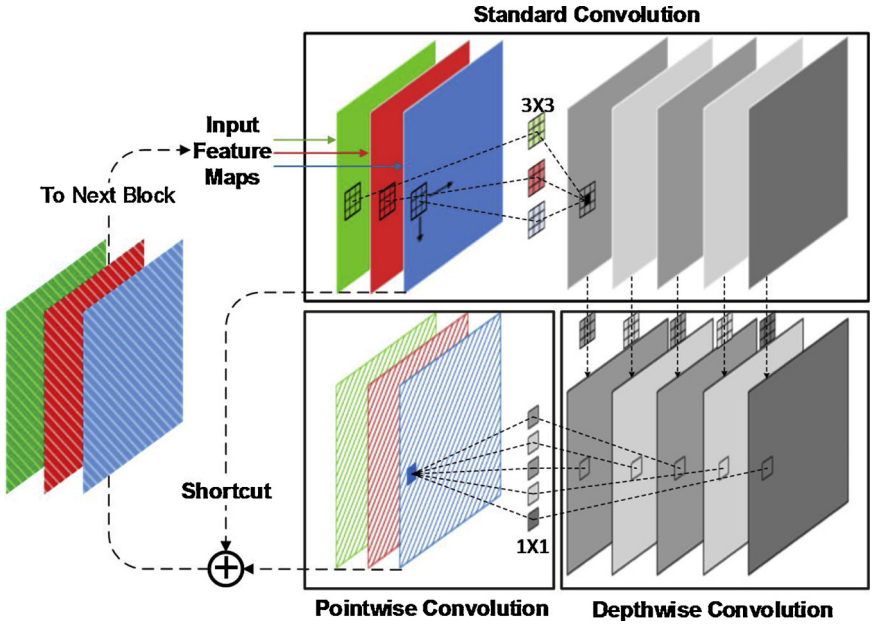

Fig. 4. Illustration of the inverted residual block structure.

output of pointwise convolution is not activated by activation functions to accommodate information preservation, which has been well discussed in [34].

The average vector **X** of the final produced feature maps is then processed by a fully connected layer with Softmax activation. The ultimate output of the image identification module should be a vector like $[p_1, p_2,\ldots, p_n]$, where p_i denotes to the exist probability of pattern i, and is calculated as follows:

$$p_i = e^{\mathbf{W}_i\mathbf{X}+b_i} / \sum_{j=1}^{J} e^{\mathbf{W}_j\mathbf{X}+b_j} \tag{8}$$

where J denotes to the number of patterns. $\boldsymbol{W}$ and $\boldsymbol{b}$ are a weight vector and bias, respectively. The cross-entropy loss function is used to measure the error between the output and the target label, which is expressed as:

$$L = -\sum_{j=1}^{J} y_i \ln(p_i) \tag{9}$$

The calculated error value is then backpropagated to train the uncertain parameters in the kernels and weight matrices. The overall architecture of the neural network is presented as shown in Table 1. The size of the input image in each layer is calculated based on a width multiplier of 0.5. H, W and C denote the height, width and number of channel respectively. k denotes the size of convolutional kernel (except kernels used in pointwise convolutions), s denotes the step of convolutional operation. Each block is repeated by cascading n times.

Table 1
The architectural structure of proposed Neural network.

Input($H \times W \times C$)	Block Type	n	k	s
120 × 50 × 3	Standard Convolutional	–	3	2
60 × 25 × 32	Inverted residual block	1	3	1
60 × 25 × 8	Inverted residual block	1	3	2
30 × 13 × 16	Inverted residual block	2	3	2
15 × 7 × 32	Inverted residual block	3	3	1
15 × 7 × 64	Inverted residual block	2	3	1
15 × 7 × 128	Standard Convolutional	–	1	1
15 × 7 × 256	Global average pooling layer	–	–	–
256	Fully Connected layer	–	–	–

4. Model evaluation

4.1. Experiment preparation

An experiment study was conducted to evaluate the proposed product inspection method in this section. Foreign matters and contaminations were introduced to simulate improper manufacturing process. These images were labeled as several different categories to correspond to the actual type of the source bottle. In summary, the dataset provides 332 images which are divided into four categories, including bottles with normal product, bottles with small foreign matter, bottles with large foreign matter and contaminated bottles. We used 70% of the image dataset for training and 30% for testing. Some defective bottle samples are shown in Fig. 5, where images are of size 1286 × 962. It is obvious that the positions of the bottles shift during the capturing process, which means it may not be effective to extract ROI from raw images using a rectangular area with a fixed size and coordinate.

The proposed inspection method was tested on the dataset for ten times, and the dataset was repartitioned into training set and testing set in each trial. The training set was used to optimize the parameters in the neural network; then the trained model was evaluated on the testing set by serval performance metrics. Specifically, the averaging inspection accuracy and recall in the ten trails were employed to measure the accurateness of the proposed inspection system. Besides, we use the averaging prediction time consumption on one image sample to measure the processing speed and efficacy of the system. The aforesaid performance metrics are defined as follows:

$$\text{Accuracy} = \sum_{i=1}^{n} T_i / \left(\sum_{i=1}^{n} T_i + \sum_{i=1}^{n} F_i \right) \tag{10}$$

$$\text{Recall} = T_i / (T_i + F_i) \tag{11}$$

where T_i denotes to the number of successfully inspected bottles with type i, and F_i denotes to the number of misclassified bottles with type i. n is the number of categories which equals to four in our case.

The image processing and identification modules were programmed based on the machine vision library OpenCV [38] and the deep learning framework Tensorflow [39]. A personal machine (CPU: Intel Xeon W-2102; GPU: Quadro P600-2G; RAM: 8G) is used for running the program.

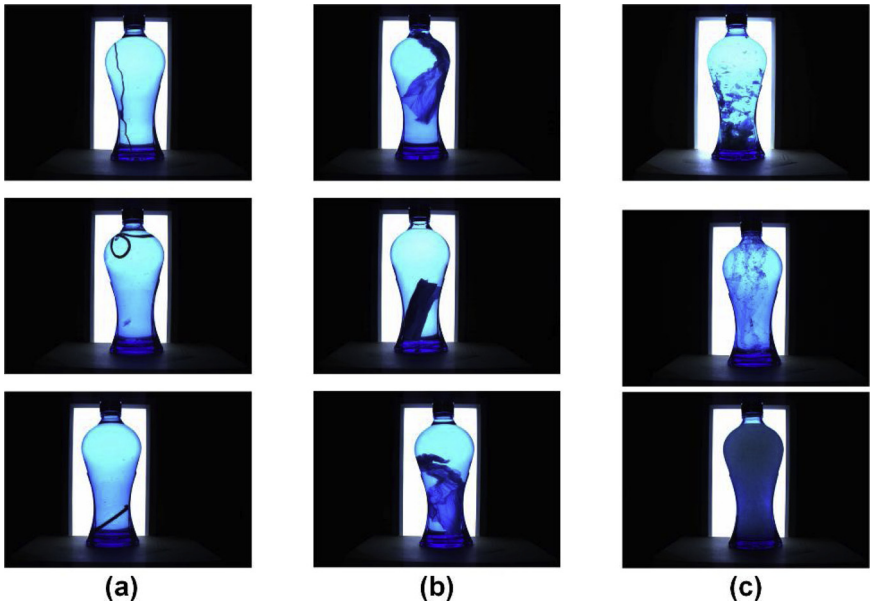

Fig. 5. Defect types. a) Small Foreign matter, b) large foreign matter, c) contamination.

4.2. Hyperparameter setup for the neural network

Training a favorable neural network heavily depends on the setup of hyperparameters. This section presents some of the key hyperparameter options in the aspects of network architecture and training strategy.

For the network architecture, the size and number of the convolutional kernels are essential, which are highly related to the quality of extract image features. Except for point-wise convolutional layers, all convolutional kernels have a size of 3×3 which is the smallest convolution kernel that can perceive local spatial information. As stated in [40], small convolutional kernels replace large convolutional kernels not only to bring performance improvements through regularization, but also to reduce the storage of the network. On the other hand, the number of convolutional kernels is actually determined by the number of output channels and convolutional type (standard or deepwise). A large number of output channels enable the layer to get more kinds of local features from the input feature maps, and then obtain rich high-level features by combining them nonlinearly. However, this is usually accompanied by the risks of computational burden increasing and overfitting. In this study, experiments were initially performed when the number of output channels in each inverted residual block equals to 16, 32, 64, 128 and 256, respectively, and width multiplier α is introduced to rapidly correct the number of filters, which is set to 0.5 finally according to the parameter analysis presented in the later of this section.

In the training phase, a challenging problem is how to determine the learning rate which controls how much the weights and biases will be adjusted according to the error. Conventional stochastic gradient descent optimization algorithm with an improper learning rate settings may lead to a slower convergence or diverge. Adam optimizer [41], which calculates independent adaptive learning rates for different parameters through the first-order moment and second-order moment estimation of the gradient, is introduced for network training. Although the optimizer itself needs to be configured, it is believed that Adam can train a willing network in a quite large range. In summary, we follow the optimizer settings recommended in [41], where the learning rate is initialized as 0.001. A mini-batch size is another important hyperparameter in the training phase, which refers to the number of samples that are fed into the network at a time. Empowered by the parallel mechanism in GPU, a large mini-batch size could effectively accelerate the overall training process, but also lead to a large memory footprint. In our case, the batch size is set to four to make full use of computing device and memory space.

4.3. Experiment results

After weights and biases initialization, the neural network was trained for 50 loops through the cropped and compressed images from training dataset, and the overall accuracy on testing dataset is 99.60%. Classification results are detailed in Fig. 6 as a confusion matrix, and the values in the diagonal of the graph indicate the recall of the inspection result on one specific bottle type. The result demonstrates that our inspection method extracts ROI from raw captured image correctly and achieves bottle type classification with high accuracy even with the influence of slight image shifting. The favorable inspection accurateness can be explained in two aspects: 1) with the assistant of a rectangular light source, probabilistic Hough transform based ROI extraction method can get the correct bottle region in the original image; 2) as a deep learning model, the constructed neural network inherits the capability of essential features extraction from natural data, thus achieving effective identification. With the small number of available samples in mind, although a few samples are misclassified, our methods have the potential to obtain more accurate results as more defective samples are discovered during production and used for network training.

The time share of each processing stage is illustrated in the top of Fig. 7. The proposed inspection method spends 47.60 ms on each image on average. In other words, a server installed with the proposed programs is capable of inspecting 21 products per second, which is sufficient for most production lines. Specifically, the time consumption of image identification module is 35.8 ms, much higher than Gaussian filtering, graying, edge detection, line detection, image clipping and compression, which are 2.66 ms, 1.12 ms, 2.38 ms, 2.39 ms, 1.69 ms and 1.58 ms, respectively. These gaps become smaller when the GPU device is enabled to parallel process the captured images within neural network, where each image merely requires 19.40 ms to be identified, which makes a deep learning based real-time inspection system more practical. The bottom of Fig. 7 presents the timeline of the proposed method without ROI extraction module. We keep the Gaussian filtering step and the image compressing step for a fair comparison, and the result clearly demonstrates the significance of ROI extraction. Since a large number of meaningless background contents are taken into consideration by neural network, the size of feature maps in each layer also

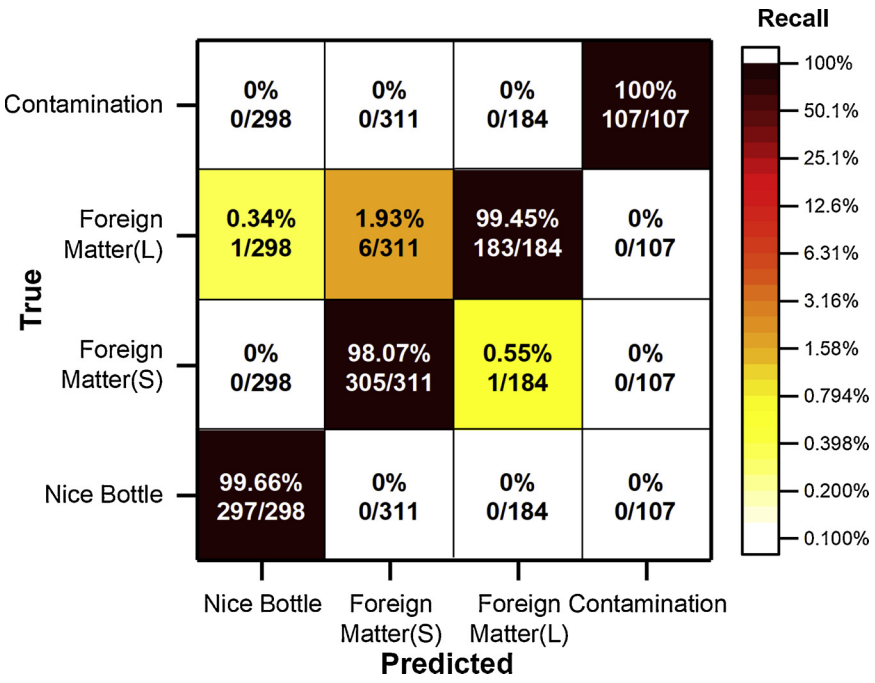

Fig. 6. Confusion matrix of bottle defect classification.

grows, thereby resulting in more unnecessary calculations. Although the proposed method requires more processing steps, the computational complexity of these image processing steps is negligible compared to the identification module.

To give an intuitive understanding of the hierarchical feature extraction process in the neural network, feature maps generated by serval intermediate layers are illustrated along with their low-dimensional representations, as shown in Fig. 8. Specifically, processed dataset is fed to the trained neural network; then the feature maps or feature vectors are derived from the first convolutional layer, the 4th and the 8th inverted residual blocks and the global averaging layer. A nonlinear dimensionality reduction method, named t-SNE [42], is utilized to map these high dimensional matrices and vectors to a two-dimensional map, and we use different colors to indicating different bottle types in the two-dimensional map. From Fig. 8, feature maps in the lower layer are similar to edge maps, however, with the layer going deeper, the feature map becomes more and more abstract and concise, which confirms that the neural network automatically learns to capture important visual features in a supervised way. As for the two-dimensional map, it is obvious that low-dimensional representations are heavily overlapped at the input layer, whereas they become more separable in the deeper layer. In the last two-dimensional map, almost all samples are clustered to correct clusters, making it much easier for the final classification. The feature visualization project is important to correctly interpret the strong image recognition capabilities of the neural network.

4.4. Identification performance comparison

The developed image identification method is not only compared with traditional feature based methods employing shallow machine learning algorithms to verify the capability of discriminative feature learning, but also with the state-of-the-art deep learning-based methods to show the enhanced processing speed. These comparable methods are detailed as follows:

- BOW + SVM: the BOW feature extractor is built based on Histogram of Oriented Gradient feature (HOG), and the number of words is set to 72. Next, the extracted features are classified by a support vector machine (SVM) which employs histogram intersection kernel, and the complexity factor and tolerance factor are 1 and 0.01, respectively.

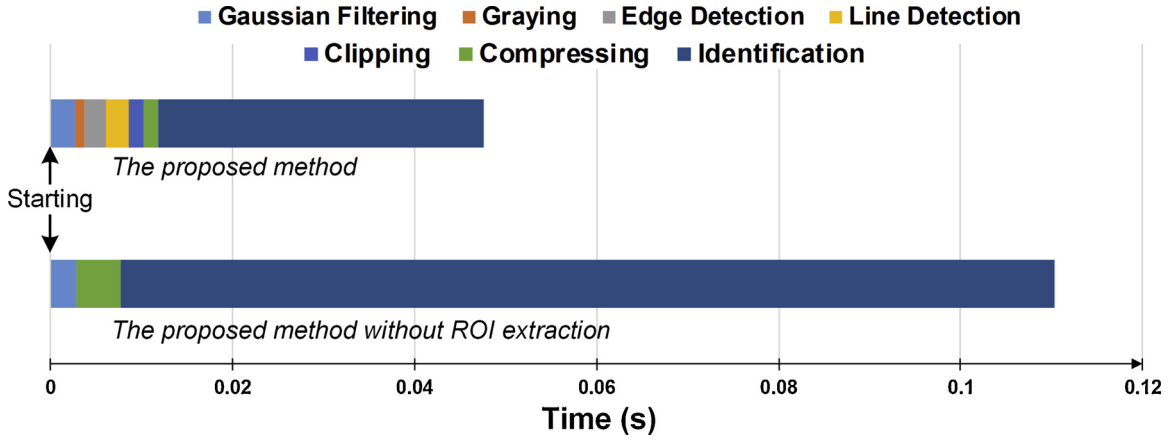

Fig. 7. Time consumption analysis.

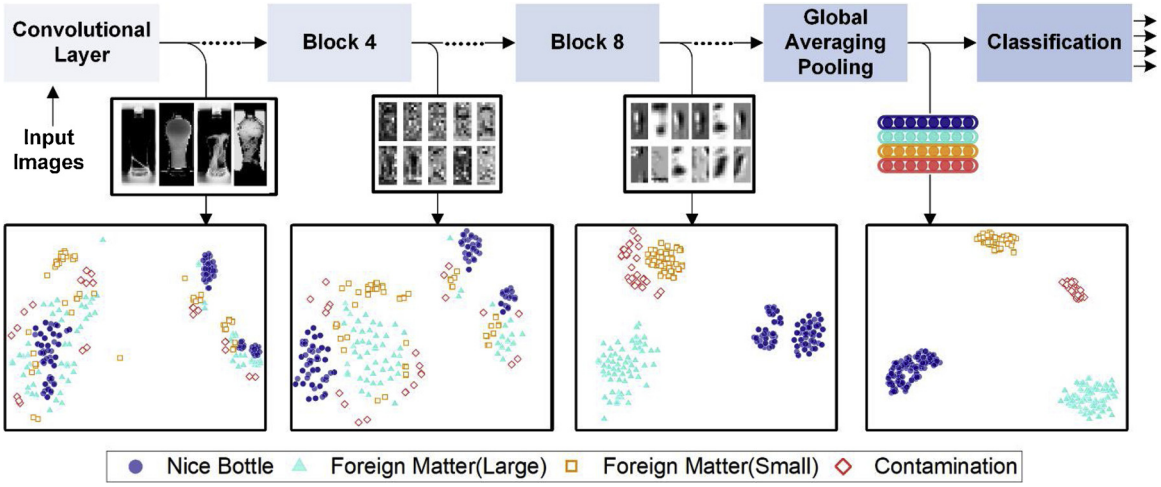

Fig. 8. Visualization of the extracted features.

- MLP: A multilayer perceptron (MLP) with a size of [128, 64, 4] is used to classify flattened and normalized image matrix. The first two layers employ Relu as the activation function and the last layer is a Softmax classification layer. This method employs the same training strategy as the proposed method.
- CAE + SVM: A convolutional auto-encoder (CAE) is first performed to get efficient image features in an unsupervised manner. The convolution layer size and kernel size are defined as [256, 128, 64] and three, respectively. After each convolution layer, a max pooling layer is used for down sampling. Based on extracted features, a SVM with a linear kernel is utilized for classification.
- Tiny ResNet: This model has a similar structure to the proposed model, where the deepwise convolutions are replaced by standard convolutions, and the rest of the network remains unchanged. Since the short-cut connection are remained, the network could be regard as a tiny version of residual network.

The comparative experimental results are shown in Table 2. It is noted that all of the evaluated methods use well-preprocessed images. From the results, it is obvious that most deep learning methods other than MLP require more computational resource to yield better inspection accuracy. MLP is the fastest but also the least accurate as MLP is fail to capture spatial information in the image. Meanwhile, since MLP is parameter intensive, a larger space is required to cache and store the trained MLP, which may result in an incompatibility with embedded devices in practice. In contrast, benefiting from the convolutional kernels with shared weight, other deep learning methods effectively reduce the model size while achieving higher accuracy. Even with a simpler architecture and less parameters, the developed neural network still performs slightly better than the CAE + SVM and tiny Resnet which employs standard convolution operation rather than deepwise version. Besides the higher classification accuracy, the processing speed and model size of the developed neural network are also much less, especially the latter one.

Briefly, based on the comparisons of experimental results using different methods, we can observe that the proposed method is able to recognize the defective products accurately and quickly, which can mainly be attributed to the utilization of inverted residual blocks.

Table 2
The comparative results of different methods.

Methods	Accuracy	Size of Model	Time Consumption
BOW + SVM	93.07%	2421 + 87 kB	11.50 ms
MLP	78.20%	35,678 kB	5.97 ms
CAE + SVM	98.60%	5233 + 12,169 kB	40.29 ms
Tiny ResNet	98.90%	16,141 kB	46.63 ms
Proposed neural network	99.60%	1176 kB	35.80 ms

4.5. Parameter analysis

Recalling to Eq. (7), width multiplier α is introduced to control the number of output channels in each inverted residual block, thus enabling designers to conveniently shrink ($\alpha < 1$) or expand ($\alpha > 1$) the overall size of the neural network after a baseline architecture has already been built. It is clear that a small width multiplier will accelerate the identification process, but unavoidably decreasing the identification accuracy. With a large width multiplier, the neural network may work better on the training images, but more computational burden will be introduced, and the testing images are more likely to be misclassified due to overparameterization. To verify the above statement empirically, ten kinds of model compression cases (width multiplier) are used to investigate the trade off between identification accuracy and time consumption, as displayed in Fig. 9, and the results are in align with our assumption. As the width multiplier grows, time consumption increases linearly. Under a moderate width multiplier, our identification module is able to achieve optimal accuracy without overfitting or underfitting the training images.

5. Conclusions

This paper presents a machine vision based inspection method for product quality control, which employs deep learning model as the core of image identification. To overcome the problem that deep learning model based image analyzing is typically computational expensive, the proposed method not only integrates conventional image processing method to obtain the ROI to remove the task-unrelated background content, but also introduces inverted residual block as the basic building block of the neural network. By replacing the standard convolution to depthwise convolution and directly linking the input and output, inverted residual block could effectively reduce the model size and offload computational burden without loss of inspection accuracy. The experimental study on defective bottles inspection demonstrates the usefulness of the proposed method.

In the future work, we will explore the combination of the proposed method with other machine learning techniques. Most of the exist learning based inspection methods are trained with the entire training data which may not be available prior to the task. Moreover, once the learning model has been built, it is hard to well utilize the newly obtained defective samples. Online machine learning provides a paradigm that allows model to be updated and optimized at each identification

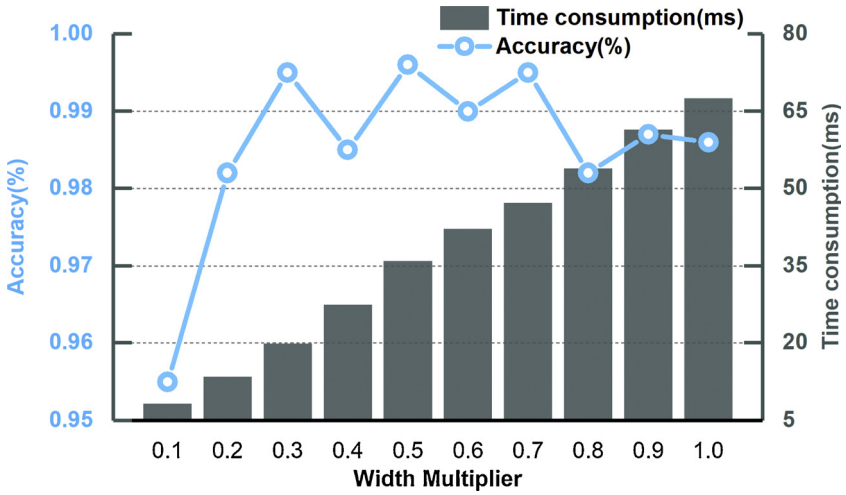

Fig. 9. Indentification performances of our neural network under different width multipliers.

step, which is a promising way to handle manufacturing data stream.

Acknowledgments

This research acknowledges the financial support partially provided by Natural Science Foundation of China (No. U1862104), National Key Research and Development Program of China (No. 2016YFC0802103) and Science Foundation of China University of Petroleum, Beijing (No. ZX20180008).

References

[1] Wang J, Ma Y, Zhang L, Gao RX, Wu D. Deep learning for smart manufacturing: methods and applications. J Manuf Syst 2018;48:144–56.
[2] Luo WC, Hu TL, Zhang CR, Wei YL. Digital twin for CNC machine tool: modeling and using strategy. J Ambient Intell Humaniz Comput 2019;10(3):1129–40.
[3] Loizou J, Tian WM, Robertson J, Camelio J. Automated wear characterization for broaching tools based on machine vision systems. J Manuf Syst 2015;37:558–63.
[4] Tootooni MS, Liu C, Roberson D, Donovan R, Rao PK, Kong Z, et al. Online non-contact surface finish measurement in machining using graph theory-based image analysis. J Manuf Syst 2016;41:266–76.
[5] Fernandez-Robles L, Azzopardi G, Alegre E, Petkov N, Castejon-Limas M. Identification of milling inserts in situ based on a versatile machine vision system. J Manuf Syst 2017;45:48–57.
[6] Wang TJ, Kwok TH, Zhou C, Vader S. In-situ droplet inspection and closed-loop control system using machine learning for liquid metal jet printing. J Manuf Syst 2018;47:83–92.
[7] Schmidt B, Wang LH. Depth camera based collision avoidance via active robot control. J Manuf Syst 2014;33:711–8.
[8] Kuo CFJ, Lo WC, Huang YR, Tsai HY, Lee CL, Wu HC. Automated defect inspection system for CMOS image sensor with micro multi-layer non-spherical lens module. J Manuf Syst 2017;45:248–59.
[9] Aminzadeh M, Kurfess T. Automatic thresholding for defect detection by background histogram mode extents. J Manuf Syst 2015;37:83–92.
[10] Malamas EN, Petrakis EGM, Zervakis M, Petit L, Legat JD. A survey on industrial vision systems, applications and tools. Image Vis Comput 2003;21(2):171–88.
[11] Brosnan T, Sun DW. Improving quality inspection of food products by computer vision – a review. J Food Eng 2004;61(1):3–16.
[12] Li D, Dong Y. Deep learning: methods and applications. Found. Trends Signal Process. 2013;7(3–4):197–387.
[13] Jason S, Edward K. CUDA by example: an introduction to general-purpose GPU programming. 1st ed. Boston: Addison-Wesley; 2010.
[14] Li LZ, Ota K, Dong MX. Deep learning for smart industry: efficient manufacture inspection system with fog computing. IEEE Trans Ind Inform 2018;14(10):4665–73.
[15] Wu D, Liu S, Zhang L, Terpenny J, Gao RX, Kurfess T, et al. A fog computing-based framework for process monitoring and prognosis in cyber-manufacturing. J Manuf Syst 2017;43:25–34.
[16] Iglesias C, Martinez J, Taboada J. Automated vision system for quality inspection of slate slabs. Comput Ind 2018;99:119–29.
[17] Haddad BM, Yang S, Karam LJ, Ye JP, Patel NS, Braun MW. Multifeature, sparse-based approach for defects detection and classification in semiconductor units. IEEE Trans Autom Sci Eng 2018;15(1):145–59.
[18] Zeng JL, Chang BH, Du D, Hong YX, Zou YR, Chang SH. A visual weld edge recognition method based on light and shadow feature construction using directional lighting. J Manuf Process 2016;24:19–30.
[19] Roberto B. Template matching techniques in computer vision: theory and practice. 1st ed. Hoboken: Wiley; 2009.
[20] Tsai D, Hsieh Y. Machine vision-based positioning and inspection using expectation-maximization technique. IEEE Trans Instrum Meas 2017;66(11):2858–68.
[21] Kong H, Yang J, Chen Z. Accurate and efficient inspection of speckle and scratch defects on surfaces of planar products. IEEE Trans Ind Inform 2017;13(4):1855–65.
[22] Jian C, Gao J, Ao Y. Automatic surface defect detection for mobile phone screen glass based on machine vision. Appl Soft Comput 2017;52:348–58.
[23] Tao X, Wang ZH, Zhang ZT, Zhang DP, Xu D, Gong XY, et al. Wire defect recognition of spring-wire socket using multitask convolutional neural networks. IEEE Trans Compon Packag Manuf Technol 2018;8(4):689–98.
[24] Tello G, Al-Jarrah OY, Yoo PD, Al-Hammadi Y, Muhaidat S, Lee U. Deep-structured machine learning model for the recognition of mixed-defect patterns in semiconductor fabrication processes. IEEE Trans Semicond Manuf 2018;31(2):315–22.
[25] Yang YX, Lou YT, Gao MY, Ma GJ. An automatic aperture detection system for LED cup based on machine vision. Multimed Tools Appl 2018;77(18):23227–44.
[26] Yuan ZC, Zhang ZT, Su H, Zhang L, Shen F, Zhang F. Vision-based defect detection for mobile phone cover glass using deep neural networks. Int J Precis Eng Manuf 2018;19(6):801–10.
[27] Liu LL, Yan RJ, Maruvanchery V, Kayacan E, Chen IM, Tiong LK. Transfer learning on convolutional activation feature as applied to a building quality assessment robot. Int J Adv Robot Syst 2017;14(3). https://doi.org/10.1177/1729881417712620.
[28] Mei S, Wang YD, Wen GJ. Automatic fabric defect detection with a multi-scale convolutional denoising autoencoder network model. Sensors 2018;18(4). https://doi.org/10.3390/s18041064.
[29] Khosravan N, Bagci U. Semi-supervised multi-task learning for lung cancer diagnosis. Proceedings of international engineering in medicine and biology conference. 2018. p. 710–3.
[30] Shao CH, Paynabar K, Kim TH, Jin JH, Hu SJ, Spicer JP, et al. Feature selection for manufacturing process monitoring using cross-validation. J Manuf Syst 2013;32(4):550–5.
[31] Canny J. A computational approach to edge detection. IEEE Trans Pattern Anal Mach Intell 1986;8(6):679–98.
[32] P.V.C. Hough Method and Means for Recognizing Complex Patterns. US Patents 3069654;1962.
[33] Matas J, Galambos C, Kittlerc J. Robust detection of lines using the progressive probabilistic hough transform. Comput Vis Image Underst 2000;78(1):119–37.
[34] Mark S, Andrew H, Menglong Z, Andrey Z, Liang-Chieh C. MobileNetV2: inverted residuals and linear bottlenecks. Proceedings of the IEEE conference on computer vision and pattern recognition. 2018. p. 4510–20.
[35] Ioffe S, Szegedy C. Batch normalization: accelerating deep network training by reducing internal covariate shift. Proceedings of the 32nd international conference on machine learning 2015;37:448–56.
[36] Sifre L. Rigid-motion scattering for image classification. PhD thesis. Ecole Polytechnique CMAP; 2014.
[37] Srivastava RK, Greff K, Schmidhuber J. Training Very Deep Networks. Proceedings of the neural information processing systems conference. 2015.
[38] Pulli K, Baksheev A, Kornyakov K, Eruhimov V. Real-time computer vision with OpenCV. Commun ACM 2012;55(6):61–9.
[39] Abadi M, Barham P, Chen J, Chen Z, Davis A, Dean J, et al. TensorFlow: a system for large-scale machine learning. Proceedings of the 12th USENIX symposium on operating systems design and implementation. 2016. p. 265–83.
[40] Simonyan K, Zisserman A. Very deep convolutional networks for large-scale image recognition. Proceedings of the international conference on learning representations. 2014.
[41] Kingma DP, Ba JL. Adam: a method for stochastic optimization. Proceedings of the international conference on learning representations. 2015.
[42] Van der Maaten L, Hinton G. Visualizing data using t-SNE. J Mach Learn Res 2008;9:2579–605.