



HAL
open science

Development of a Textile Coding Tag for the Traceability in Textile Supply Chain by Using Pattern Recognition and Robust Deep Learning

Kaichen Wang, Vijay Kumar, Xianyi Zeng, Ludovic Koehl, Xuyuan Tao, Yan Chen

► **To cite this version:**

Kaichen Wang, Vijay Kumar, Xianyi Zeng, Ludovic Koehl, Xuyuan Tao, et al.. Development of a Textile Coding Tag for the Traceability in Textile Supply Chain by Using Pattern Recognition and Robust Deep Learning. International Journal of Computational Intelligence Systems, 2019, International Journal of Computational Intelligence Systems, 12, pp.713-722. 10.2991/ijcis.d.190704.002 . hal-04480159

HAL Id: hal-04480159

<https://hal.univ-lille.fr/hal-04480159>

Submitted on 27 Feb 2024

HAL is a multi-disciplinary open access archive for the deposit and dissemination of scientific research documents, whether they are published or not. The documents may come from teaching and research institutions in France or abroad, or from public or private research centers.

L'archive ouverte pluridisciplinaire **HAL**, est destinée au dépôt et à la diffusion de documents scientifiques de niveau recherche, publiés ou non, émanant des établissements d'enseignement et de recherche français ou étrangers, des laboratoires publics ou privés.



Distributed under a Creative Commons Attribution - NonCommercial 4.0 International License

Development of a Textile Coding Tag for the Traceability in Textile Supply Chain by Using Pattern Recognition and Robust Deep Learning

Kaichen Wang^{1,2,*}, Vijay Kumar², Xianyi Zeng², Ludovic Koehl², Xuyuan Tao², Yan Chen¹

¹College of Textile and Clothing Engineering, Soochow University, Suzhou, 215006, China

²ENSAIT, GEMTEX, École Centrale de Lille, Lille, 59000, France

ARTICLE INFO

Article History

Received 25 Jan 2019

Accepted 24 Apr 2019

Keywords

Traceability

Textile tags

Coded yarn recognition

Deep learning

Transfer learning

Convolutional neural network

ABSTRACT

The traceability is of paramount importance and considered as a prerequisite for businesses for long-term functioning in today's global supply chain. The implementation of traceability can create visibility by the systematic recall of information related to all processes and logistics movement. The traceability coding tag consists of unique features for identification, which links the product with traceability information, plays an important part in the traceability system. In this paper, we describe an innovative technique of product component-based traceability which demonstrates that product's inherent features—extracted using deep learning—can be used as a traceability signature. This has been demonstrated on textile fabrics, where Faster region-based convolutional neural network (Faster R-CNN) has been introduced with transfer learning to provide a robust end-to-end solution for coded yarn recognition. The experimental results show that the deep learning-based algorithm is promising in coded yarn recognition, which indicates the feasibility for industrial application.

© 2019 The Authors. Published by Atlantis Press SARL.

This is an open access article distributed under the CC BY-NC 4.0 license (<http://creativecommons.org/licenses/by-nc/4.0/>).

1. INTRODUCTION

The traceability, which provides the ability to track and trace products within the supply chain, is extremely important for the manufacturing industry. The implementation of traceability enables all physical processes and information flows to be available across the manufacturing supply chain. A widely accepted definition on traceability is the “ability to verify the history, location, or application of an item by means of documented recorded identification,” which proposed by the International Organization for Standardization [1]. A better visibility and the verification of a product or raw materials within the supply chain contribute to the fight against counterfeiting [2]. The ability to track and trace enables the stakeholders to have authentic information about product distribution [3] and the origin of the identified products for recalling, which improves the efficiency of recall handling [4]. Moreover, with the traceability information related to the production process, such as the energy use, water use, the company can quantify the environmental footprint during the whole product life cycle, and further optimize the production process to reduce environmental impacts [5]. In a broad sense, an efficient traceability system enables brand owners to dynamically monitor and control all production stages of the supply chain and master key data on products and processes in order to effectively preserve professional knowledge.

As one of the oldest industries in the world, the implementation of traceability in the textile and apparel T&A industry has been proposed as novel alternatives by numerous studies [6–9]. Based on previous studies [9–11], the framework of textile supply chain traceability system is proposed, as shown in Figure 1. The traceability tag consists of unique features for identification, which links the product with traceability information, plays an important part in the traceability system [2,12]. On the business-to-business (B2B) level, industrial firms can effectively monitor data at all production stages, including technical process parameters, product quality features, and environmental impacts. With properly designed traceability tags, consumers would be able to access to the information regarding the whole production process of the final products. Besides, the information exchange between the consumer database (B2C level) and the product database can further lead to the on-demand manufacturing, which transforms manufacturing resources into services that can be comprehensively shared and circulated [13]. There are two types of tags widely used in the textile sector, namely optical tags (barcode [14], QR code [15]) and radio frequency identification (RFID) tags [16]. However, the above mentioned tags are vulnerable to replication and can be easily removed from the product without much influence on the product's physical appearance, which increases the risk of counterfeiting and the loss of tracking information. Although the RFID tags can be embedded into the product to increase the integrity with the physical product [17], they can bring some problems to the recycling and disposal of the textiles due of the use of electronic chips [18].

*Corresponding author. Email: kaichen.wang@ensait.fr

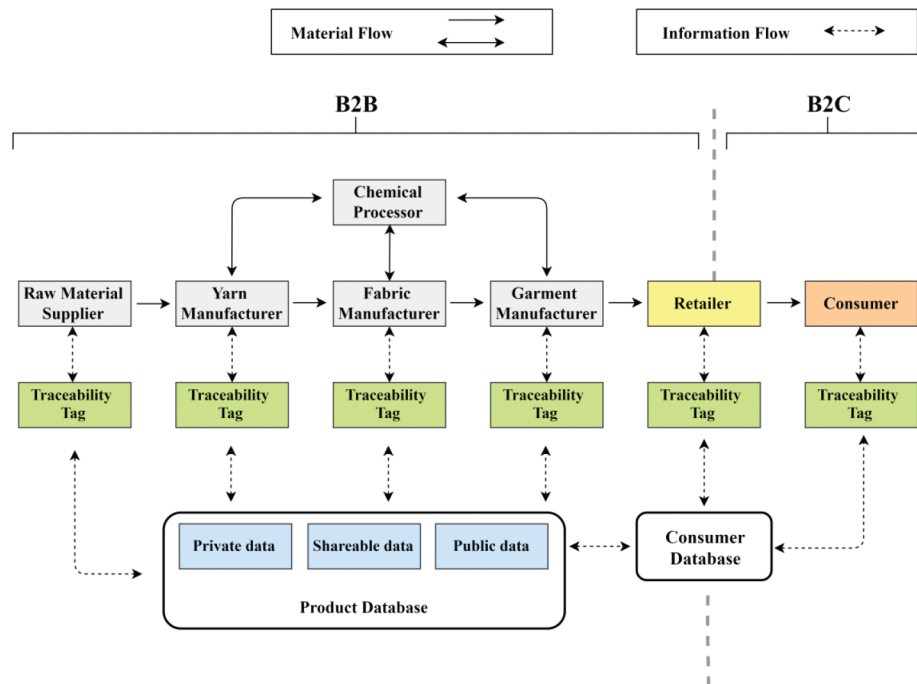


Figure 1 | The framework of textile supply chain traceability system.

In this paper, we describe an innovative technique of product component-based traceability which demonstrates that product's inherent features—extracted using deep learning—can be used as a traceability signature. This has been demonstrated on textile fabrics, where Faster region-based convolutional neural network (Faster R-CNN)-based deep learning is used to capture the visual features of textile yarns (i.e., product component) and convert it into a traceability signature. By introducing a different method for yarn coding, the number of coded yarn classes can be extended. Individual coded yarns act as the essential part of the traceability system, the sequence of coded yarn are designed accordingly to carry useful information—similar to barcode [14]. A check digit (yarn) computed by an algorithm from the other sequence input can be incorporated to detect simple errors in the input series [19]. The task of recognizing coded yarns consists of the detection of the coded yarn position—to read the sequence information, and the classification of coded yarn—to distinguish coded yarn with different constructions, which is a typical computer vision and image processing task.

The rest of the paper is organized as follows: Section 2 discusses the related work. Section 3 describes the design of the proposed coded yarn-based fabric traceability tag. Section 4 describes the object detection algorithm based on convolutional neural networks (CNNs) for tag recognition. Section 5 describes the materials used for producing the coded yarn-based tag and the method used for tag recognition. Experimental results obtained from the prepared tags are discussed in Section 6. Finally, Section 7 concludes this paper and presents the future perspectives.

2. RELATED WORK

Considering the limitation of RFIDs and printed QR codes, various propositions have been made in the past. One such proposition is, instead of using external tracking tags, the use of product's

inherent features for traceability. This not only eliminate the dependency on external tags, but the use of inherent features links the physical product with a traceability signature [6]. Further, using yarn as traceability components has been proposed for textile fabrics in which the unique optical features of the coded yarn are used as the traceability marks for the textiles [7]. With these optical features introduced by yarn coding technique, a pattern recognition algorithm is applied for decoding the information. There are several advantages offered by yarn-based tags over other tags [14,15]. The coded yarns are integrated into the textiles during the manufacturing process, thus cannot be removed from the fabric without damaging the physical appearance. Unlike the RFIDs, the coded yarns are normal textile materials, which will not affect the wearability of the product and can be recycled instead of discarded [18]. It is worth mentioning that for real world application, one of the major agitations for the textiles occurs during the washing treatment. Owing to the stressing and shearing during the washing process, the textile coding tags become flabby to a certain extent, which can slightly affect the coded yarn recognition accuracy as discussed in [8]. While for the conventional RFIDs or printed QR codes, they can no longer be functioning after washing treatment. However, there are still some limitations to the previous coded yarn-based tags. The 2-ply coded yarn consists of a core yarn and a wrapped yarn [7], sets restrictions on the diversity of the coding scheme. Besides, the pattern recognition algorithm has moderate success rates (67% and 59% for woven and knitted tags, respectively) and required the position of the reference yarn during the algorithm execution [8], which limits the application in the real production scenario.

There has been a revolution in the description of local features in the computer vision area from handcrafted to deep learning-based methods over the last decades [20]. The local feature descriptors, which describe the local regions using elements such as edges, corners, gradients, are one of the fundamental components of traditional computer vision tasks. The scale-invariant feature transform

(SIFT) algorithm using Difference of Gaussians (DoG) to detect local features proposed by Lowe [21] is one of the most widely used methods for key point detection. The data-driven methods inspired by the idea of machine learning [22] were proposed as an alternative to the selection of the best handcrafted features. These methods empower comprehensively optimized descriptors based on a specific dataset, which sometimes significantly outperform the handcrafted features [20]. In particular, multilayer artificial neural networks, also known as “deep learning” are increasing the accuracy of computer vision tasks [23]. In 2015, the deep neural network proposed by He *et al.* [24] achieved a 3.57% error rate for the correct classification of images in the ImageNet dataset [25], compared with a 5.1% error rate among human. It is worth noting that there are some works in computer vision utilizing the advantages of CNN [26–28]. Thus, instead of manually extracting coded yarn features (wrapping angle, wrapping area, and twist distance [7]), this paper utilizes the advantages of deep learning—automatically learning features are used at multiple levels of abstraction.

3. DESIGN OF CODED YARN-BASED FABRIC TAGS

3.1. Yarn Coding Procedure

The coded yarns with special optical features are used as essential component for the traceability tags. This 100% textile-based tag is a solution offered by yarn spinning process. A composite yarn assembly consists of two yarns with different contrast was proposed as the coded yarn [7]. As demonstrated in previous research [7], the optical features introduced by different twist densities of wrapping yarn and core yarn can be manually extracted for identification. In this paper, instead of using 2-ply wrapping yarn as the coded yarn, a 3-ply “Z” twisting yarn was proposed as the coded yarn. With the introducing of a third yarn in the composite yarn assembly, the number of coded yarn classes can be increased. For instance, at the same twist configuration, the 3-ply twisting yarns present different optical features with different color combination, as shown in Figure 2. Therefore, different coded yarn classes can be achieved using different twist densities together with different color combinations.

3.2. Integration of Coded Yarns into Woven Fabric Structure

Similar to the barcodes, which contains different widths of lines, the textile tags can be designed with coded yarns having different optical features. The digits in barcodes can be represented by different classes of coded yarn. The sequence of the coded yarns can be specifically designed in order to provide meaningful information. Woven structure produced by orthogonally interlacing two sets of yarns is proposed as the textile structure for coded yarn integration. As warp yarn sequence remains fixed during the weaving process, the coded yarns can be inserted on weft direction to replace some regular weft yarns. Regular yarns are inserted accordingly to separate the coded yarns. Weft selection can be easily achieved on modern weaving machines, thus the coded yarn sequence can alter according to the predesigned pattern. To introduce visible optical features, a certain length of float (long enough to extract optical features) can be used to get a continuous appearance on the

fabric surface (as shown in Figure 3). The detailed information on the production of the coded yarn-based fabric tags and the implementation is demonstrated in [8].

In order to increase the robustness of the traceability tags, a checksum algorithm, which verifies the correctness of the code sequence is integrated into the tag designing process. As shown in Figure 4, the last coded yarn in the sequence is used as the check digit (yarn) to detect whether the sequence is decoded correctly inspired by The International Article Number (also known as European Article Number or EAN) [29]. Different weights are assigned to the digits at odd and even positions, and the checksum is calculated as a weighted sum of all the digits (excluding the check digit) modulo a

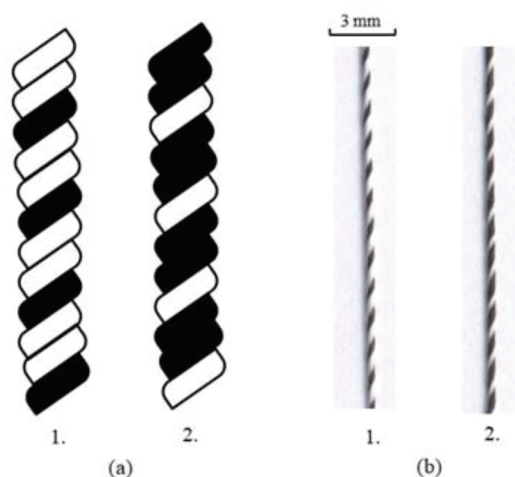


Figure 2 (a) Schematic diagram of 3-ply coded yarns, where (1) consists of 2 white yarns and 1 black yarn (2) consists of 2 black yarns and 1 white yarn; (b) Physical 3-ply coded yarn specimens, where (1) consists of 2 white yarns and 1 black yarn (2) consists of 2 black yarns and 1 white yarn.

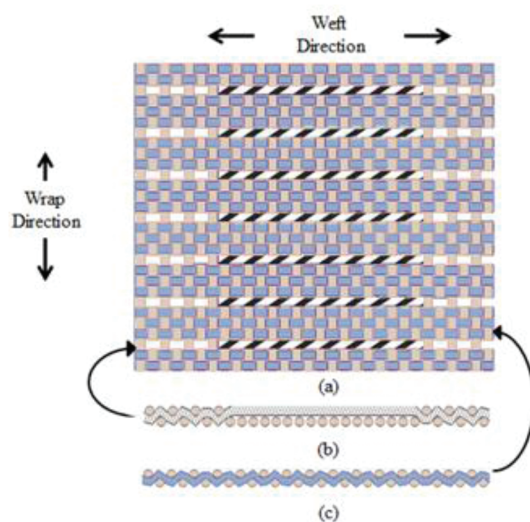


Figure 3 (a) Schematic of coded yarn-based fabric tag. Side view of coded yarn (b) and regular yarn (c) from warp direction.

specific number (depends on the number of digits used in the system). The checksum will result in a mismatch if there is only one digit incorrect in the sequence. For instance, if the digit in the first position is incorrectly detected (recognized as “2” instead of “1”), the checksum would be different because of the value altered in the first position, and the check digit would not match the calculated checksum. The actual woven tag based on a checksum designing strategy is shown in Figure 4.

4. CNN- BASED OBJECT DETECTION ALGORITHM FOR TAG RECOGNITION

The task of tag recognition is to extract the information encrypted in the tag. In our case, the tag recognition consists of the localization of the coded yarns and the classification of the coded yarns simultaneously based on the special optical features of the coded yarns. The information integrated into the tag can be extracted once the sequence of the coded yarns is detected and verified by the checksum algorithm.

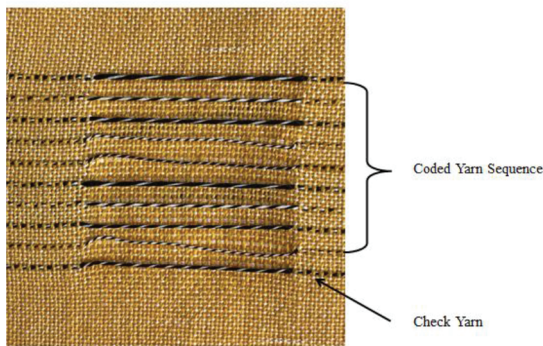


Figure 4 | Physical woven fabric tag specimen based on a checksum designing strategy.

The traditional approaches such as HOG [30], SIFT [21], and SURF [31], which heavily dependent on the manually extracted features are not robust regarding the illumination conditions, image scale changes and image rotations. The flexible nature leads to the distortion and bending of the coded yarn in textiles, which brings a variation on the optical features. Membership functions based on the optical feature distribution are used for the classification of coded yarns, as discussed in [7]. However, though the extracted optical features can be successfully incorporated with yarn classification, the manually selected features and parameters diminished the robustness when there is a certain variety in characteristics of the coded yarns.

Therefore, a CNNs based method—a data-driven approach, is used in this paper for the tag recognition. The fundamental idea of CNNs founded on the application of multiscale filters to achieve suitable features with the raw data with minimal preprocessing [32]. The basic features (colors, edges) are delivered by the preliminary convolutional blocks and propagate through the whole network structure, which enabled a sophisticated representation for the final classification [33]. In this paper, considering the aforementioned reasons, Faster R-CNN [34] is adopted for the tag recognition. The framework of the coded yarn recognition system is shown in Figure 5.

4.1. The Network Architecture of Faster R-CNN

In 2014, Girshick *et al.* [35] proposed the idea of combining region proposals with CNNs, which used Selective Search algorithms to propose Regions of Interests (RoI) and further a CNN for classification and adjustment. However, the training process is tedious because the two parts of the network cannot be trained at the same

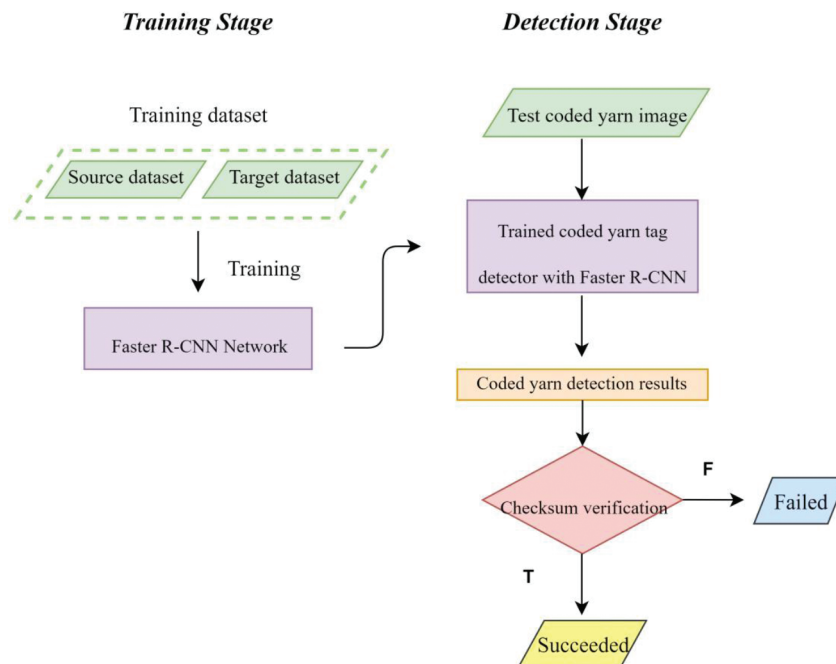


Figure 5 | The framework of coded yarn-based fabric tag recognition system with Faster region-based convolutional neural network (Faster R-CNN).

time. In order to improve the training process, Fast R-CNN [36] was published in 2015, where the Region of Interest Pooling algorithm was used to replace the Selective Search. Faster R-CNN [34] further improves the training speed and accuracy by combining the features of a fully differentiable convolutional network to do both region proposals and detection. The whole framework of Faster R-CNN consists of two modules, namely the region proposal network (RPN) and the Fast R-CNN detector, as shown in Figure 6. The RPN is a fully convolutional network, which effectively generates region proposals with multiple scales and aspect ratios. The Fast R-CNN detector is then used to refine the proposed regions. The two modules share the same convolutional layers is utilized for joint training, which significantly increases the training speed and enables the framework to train deep networks [34].

4.2. CNN Extractor

A CNN block is initially used as the feature extractor for input images. However, deep learning models have a large number of parameters to be trained, which often require a large dataset [37]. Transfer learning [38] was proposed to improve the model performance by avoiding tedious data-labeling. In this process, a CNN trained on a large source dataset was used as a base model, followed by a fine-tuning according to a small customized target dataset. As we start training with a trained model, fewer data is required for the training a deep neural network. Fine-tuning is feasible as the target dataset and the source dataset has some overlap on the sample space [39]. In this paper, we use the PASCAL VOC [40] dataset as the source dataset and fine-tune the ResNet-101 [24] model on our target dataset, which consists of annotated images for our coded yarn-based fabric tags. In our previous research work, some optical features of the coded yarn are manually extracted for tag recognition; while in this paper, the features are automatically extracted by the CNN network at multiple levels of abstraction [33].

4.3. Region Proposal Network

The RPN takes the previous convolutional feature map as input, and outputs a series of rectangular object proposals (coded yarns) with scores of objectness accordingly. The RPN slides over the convolutional feature map proposed by the last shared convolutional layer

to generate region proposals. The slided features based on the convolutional computation with feature map are then imported to two fully 1 by 1 convolutional layers for box regression and box classification, respectively. For the box regression layer, the coordinates of the bounding boxes were generated. While for the box classification layer, objectness scores were proposed based on whether the proposals contains an object (coded yarns). Since highly overlapping occurred among the initial proposals, proposals with high Intersection over Union (IoU) are merged using nonmaximum suppression (NMS) [41]. In the RPN training process, if the proposal has an IoU overlap higher than a threshold (0.7 in [34]) with the ground-truth box, we assign the proposal to be a positive label; if the proposal has an IoU overlap lower than a threshold (0.3 in [29]) with the ground-truth box, we assign the proposal to be a negative label. It is worth mentioning that the proposals are associated with multiply scales and aspect ratios, which provide the translation-invariant properties over other methods [34]. Following the multi-task loss in Fast R-CNN [36], the loss function is defined in Equation (1):

$$L(\{p_i\}, \{t_i\}) = \frac{1}{N_{cls}} \sum_i L_{cls}(p_i, p_i^*) + \lambda \frac{1}{N_{reg}} \sum_i p_i^* L_{reg}(t_i, t_i^*) \quad (1)$$

The multi-task loss has two parts, a classification term L_{cls} and a regression term L_{reg} . i is the index of an anchor (proposal) and p_i is the predicted probability of anchor i being an object. p_i^* represents the ground-truth label, where $p_i^* = 1$ if the anchor is positive and $p_i^* = 0$ if the anchor is negative. t_i is a vector representing the four parameterized coordinates of the predicted bounding box; and t_i^* is the vector of the ground-truth box associated with a positive anchor. Softmax loss is used for the classification loss L_{cls} , while smooth L1 [36] is used for regression loss L_{reg} . The above two terms are normalized by N_{cls} and N_{reg} , respectively, and λ (set as 10 [34]) is used as a balancing weight.

4.4. Classification Network

Several RoIs are proposed by the RPN. For each RoI, the RoI pooling layer from the convolutional layers is used to extract a fixed length feature vector. Then each feature vector is fed to a sequence of fully connected layers. Finally, a Softmax loss is used as a classifier

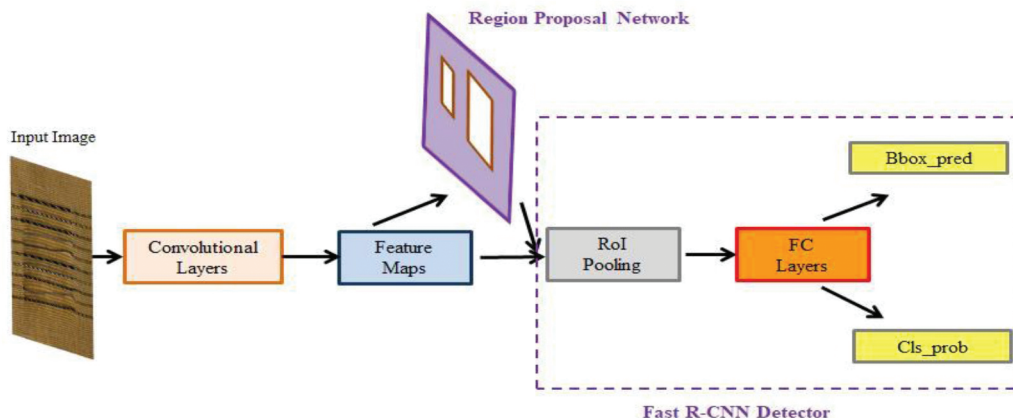


Figure 6 | The network architecture of Faster region-based convolutional neural network (Faster R-CNN).

to distinguish the object classes and the background class. The values of the bounding-box coordinates are further optimized with the related regression layer. Thus, after training, the whole network can gain the ability to locate the coded yarn and distinguish the coded yarn class at the same time.

5. MATERIALS AND EXPERIMENTS

The procedure described in Section 2.1 was used to create different optical features for the coded yarns. As shown in Figure 7, six coded yarn classes based on different twists and color sequences were produced using 8.7 Tex¹ black-and-white polyester yarns. Each type of the coded yarn represents a digit, which can be further designed to generate meaningful information. The specifications of the coded yarns are listed in Table 1. Further, woven fabric tags were produced following the procedure discussed in Section 2.2. The aforementioned coded yarns were integrated into the woven fabrics in predefined sequences during the weaving process, and the last coded yarn in a specific sequence was designed as the check digit (yarn) based on the checksum algorithm as illustrated in Section 2.2. The maximum repeat pattern in a coded yarn is around 5 mm (at 200 TPM²), thus the float length is selected to be around 27 mm in order to have at least four complete patterns for feature extraction. Since hierarchical pooling layer are used in the CNN blocks for the coded yarn detection, the multi-scale features can be extracted without reference scale and predefined position of the coded yarn as indicated in [8], which enables an efficient end-to-end solution.



Figure 7 | Different classes of coded yarn specimens, where the annotated number indicates the digit represented by each coded yarn class.

Table 1 | Specifications of coded yarn classes.

| Represented digit | 0 | | 1 | | 2 | | 3 | | 4 | | 5 | |
|-------------------------|---------|---------|---------|---------|---------|---------|---------|---------|---------|---------|---------|---------|
| No. of twists per meter | 200 | | 200 | | 400 | | 400 | | 800 | | 800 | |
| Color composition | 2 White | 1 Black | 1 White | 2 Black | 2 White | 1 Black | 1 White | 2 Black | 2 White | 1 Black | 1 White | 2 Black |

¹Tex: gram per kilometer

²TPM: twists per meter

In this paper, 54 coded yarn-based woven fabric tags were produced for the analysis, where 40 tags were used for training, 5 tags were used for validation, and 9 tags were used for testing. For each tag, two 1500 × 1500 pixels images of were captured using a digital camera under different day-light conditions (bright sunlight, illuminated shade, etc.) and different image capturing distances (from 20 cm to 40 cm) in order to increase variance in our image dataset. The captured images are at 96 dots per inch (DPI), which is high enough to express the optical features of individual coded yarns. The specifications of our total 108 image dataset and the contained coded yarns are listed in Table 2. As can be seen from Table 2, the distribution of different classes of coded yarns is roughly the same for training, validation, and testing sets. Since the deep learning-based recognition system performs regression on the bounding-box positions of coded yarns during the training process, no preprocessing processes were applied to the captured images.

It's worth mentioning that different noises can be introduced during the image capturing and transmitting process. Therefore, two different types of noises commonly occurring in image capture, zero-mean Gaussian (ZMG) noise (additive noise), and salt-and-pepper noise (S&P), also known as impulse noise, were synthetically added to the testing images. ZMG noise was introduced randomly at four variance levels, namely 0.01, 0.02, 0.05, and 0.10. S&P noise was introduced at four levels, namely 1%, 2%, 5%, and 10%. For x% of S&P noise, the salt noise and the pepper noise were equally introduced, which means each accounted for half of the x% noise. In this paper, the tag recognition algorithm was implemented in PyTorch [42], and the Faster R-CNN framework is based on [43]. The proposed algorithm was run on a Linux machine (Ubuntu 16.04) with Intel Core i5 2.5 GHz and 8GB RAM. A 12GB NVIDIA Titan X GPU was used for training.

The recognition of coded yarns is a multi-object detection task, thus mean average precision (mAP) [40] is used to evaluate the performance. The mAP is the mean value of the average precisions (APs) of each object classes (coded yarns), where AP summarizes the precision/recall curve. The AP is defined as the mean precision at a set of 11 equally spaced recall levels [0, 0.1, 0.2, ..., 1], where precision at each recall level is interpolated by taking the maximum precision measured for which the corresponding recall exceeds the certain level, as discussed in detail in [40]. The precision and recall for each object classes (coded yarns) are defined in Equations (2) and (3).

$$Precision = \frac{TP}{TP + FP} \quad (2)$$

$$Recall = \frac{TP}{TP + FN} \quad (3)$$

where TP is the number of True Positive objects—the correctly detected coded yarns; FP is the number of False Positive objects—incorrectly detected coded yarns, and FN is the number of False Negative objects—undetected coded yarns. It is noteworthy that the decision of Positive/Negative in object detection is based on

Table 2 | Specifications of image datasets.

| | No. of Images | No. of Different Classes of Coded Yarns | | | | | |
|------------|---------------|---|-----|-----|-----|-----|-----|
| | | 0 | 1 | 2 | 3 | 4 | 5 |
| Training | 80 | 136 | 128 | 144 | 136 | 136 | 120 |
| Validation | 10 | 16 | 20 | 22 | 18 | 12 | 12 |
| Testing | 18 | 32 | 28 | 24 | 26 | 32 | 38 |
| Total | 108 | 184 | 176 | 190 | 180 | 180 | 170 |

Table 3 | Comparison of coded yarn -based fabric tags detection results on testing sets with and without noises.

| Testing Set | Average Precision of Different Classes of Coded Yarns (%) | | | | | | mAP (%) |
|----------------|---|-------|-------|-------|------|------|---------|
| | 0 | 1 | 2 | 3 | 4 | 5 | |
| Without noise | 100.0 | 100.0 | 99.6 | 100.0 | 88.6 | 89.3 | 96.3 |
| ZMG noise 0.01 | 100.0 | 100.0 | 100.0 | 99.6 | 89.0 | 88.1 | 96.1 |
| ZMG noise 0.02 | 100.0 | 90.1 | 100.0 | 100.0 | 88.7 | 90.2 | 94.9 |
| ZMG noise 0.05 | 100.0 | 100.0 | 100.0 | 99.1 | 87.4 | 80.1 | 94.3 |
| ZMG noise 0.10 | 100.0 | 100.0 | 99.6 | 100.0 | 87.0 | 54.6 | 88.6 |
| S&P noise 1% | 100.0 | 90.9 | 100.0 | 100.0 | 88.7 | 89.6 | 94.8 |
| S&P noise 2% | 99.7 | 90.9 | 100.0 | 100.0 | 88.3 | 90.1 | 94.7 |
| S&P noise 5% | 100.0 | 100.0 | 90.1 | 99.3 | 88.1 | 90.9 | 94.7 |
| S&P noise 10% | 100.0 | 90.5 | 100.0 | 100.0 | 87.4 | 90.2 | 94.7 |

mAP, mean average precision; ZMG, zero-mean Gaussian; S&P, salt-and-pepper noise.

the IoU between the predicted bounding box and the ground-truth bounding box. As the purpose of our recognition task is to read the sequence of the coded yarns in the textile coding tags, the exact positions of the bounding boxes do not have a big impact on our final tag recognition results. Thus, we select a relatively low threshold value of IoU in our model [40]. The IoUs exceeded 0.5 are considered Positive instances (if the predicted label is the same the ground-truth label, this instance is considered as True Positive; if the predicted label is not the same as the ground-truth label, this instance is considered as False Positive), while the IoUs equal to 0 are considered False Negative instances.

After the recognition of the coded yarns, a sequence of digits was obtained for each tag, which was further validated by the checksum algorithm. Inspired by EAN [29], the last coded yarn in the sequence is used as the check digit (yarn) to detect whether the sequence is correct. Different weights are assigned to the digits at odd and even positions (three for even positions, one for odd positions), and the checksum is calculated as a weighted sum of all the digits (excluding the check digit) modulo a specific number. As we have six different yarn classes, we use six as the divisor. The recognition success rate is defined as the proportion of the number of successfully verified codes over the total number of codes.

6. RESULTS AND DISCUSSION

The detection results of coded yarn-based fabric tags are presented in Table 3. The detection results of testing set without noise, and testing set with different level of ZMG and S&P noise are listed accordingly. The AP for each coded yarn classes are calculated according to VOC2007 standard [40] with the mAP appended on the final column. As can be seen from Table 3, the mAP for the testing set without noise reached to 96.3%, where each of the first four classes has nearly 100% AP, and the last two classes (class “4” and “5”) have relatively lower AP (88.6% and 89.3%, respectively). With the increasing number of twists per meter (TPM), the twisting angle of the yarns increased simultaneously, while the twisting distance decreased. Moreover, the yarns with high twists tend to be unstable in the fabric. Thus, the inherent optical features of the

coded yarns may be compromised. The observation of the APs over different coded yarn classes evaluated the performance of individual class, and indicated that the higher number of TPM may result in a poorer performance, which provided guidelines for the improvement of coded yarn designing.

The testing set with 0.01 variance of ZMG noise has a mAP of 96.1%, which is nearly the same as the testing set without noise. However, as the variance level of the ZMG noise increases, the mAP drops accordingly, which indicates our tag recognition method is relatively sensitive to ZMG noise. The AP for the first four classes remains around 100%, while the AP of the last two classes is seen a slight drop with the increase of ZMG noise. As ZMG noise is applied to every pixel in the image (shown in Figure 8(a)), the noises are introduced during the convolutional operation. Thus, with the aforementioned weak inherent optical features, the class “4” and “5” coded yarns are more vulnerable to ZMG noises. Nevertheless, the mAP for testing set with 0.1 variance of ZMG noise is 88.6%, which is satisfactory for object detection task.

It is interesting to note that our tag recognition method is insensitive to S&P noises. As shown in Table 3, the mAP, as well as the AP for individual coded yarn classes remains nearly the same for all the four testing set with different level of S&P noises. Moreover, even with 10% amount of S&P noise, the mAP is 94.7%, which is close to the results for testing set with 1% S&P noise. S&P noise is one type of impulse noise applied randomly to a certain amount of pixels in the image as demonstrated in Figure 8(b). The pooling layers in the CNN structure reducing the dimension of the image can minimize the effect of randomly introduced noises and bring invariability to the network, which explains the high mAP for testing set with S&P noises. As illustrated in Figure 8, the coded yarn can still be detected under the highest level of noises introduced to the testing sets.

Table 4 shows the number of successfully and unsuccessfully verified codes, and the corresponding recognition success rate. The tag detection results were manually checked and are in consistency with the checksum verification results. The recognition success rate for testing set without noise is 94.4%, only 1 is unsuccessfully verified out of 18 codes. The recognition success rate decreases

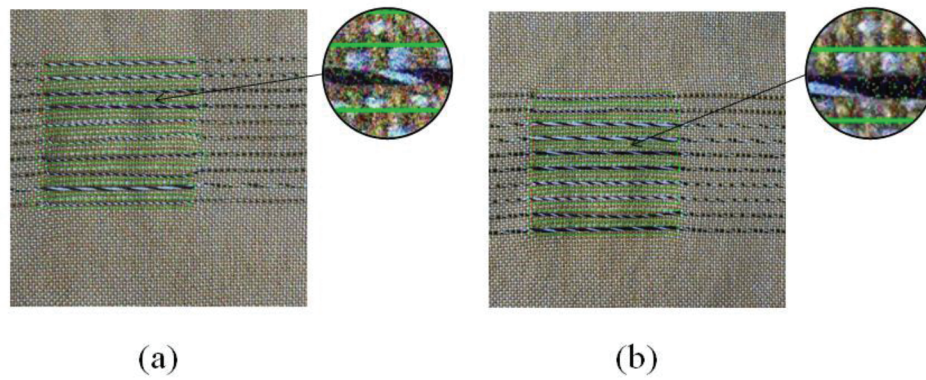


Figure 8 | Examples of detection results on testing set with different noises (a) 0.1 variance of ZMG noise (b) 10% S&P noise, the insets show specific areas at higher magnification.

Table 4 | Comparison of checksum validation results.

| Testing Set | No. of Successfully Verified Codes | No. of Unsuccessfully Verified Codes | Recognition Success Rate (%) |
|----------------|------------------------------------|--------------------------------------|------------------------------|
| Without noise | 17 | 1 | 94.4 |
| ZMG noise 0.01 | 17 | 1 | 94.4 |
| ZMG noise 0.02 | 15 | 3 | 83.3 |
| ZMG noise 0.05 | 14 | 4 | 77.8 |
| ZMG noise 0.10 | 12 | 6 | 66.7 |
| S&P noise 1% | 15 | 3 | 83.3 |
| S&P noise 2% | 15 | 3 | 83.3 |
| S&P noise 5% | 15 | 3 | 83.3 |
| S&P noise 10% | 15 | 3 | 83.3 |

ZMG, zero-mean Gaussian; S&P, salt-and-pepper noise.

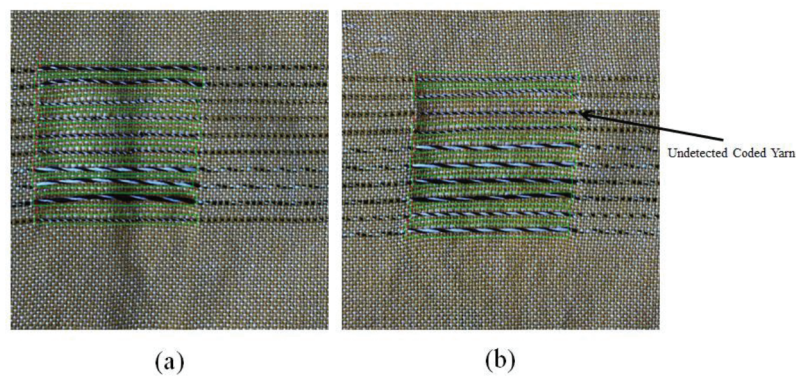


Figure 9 | Examples of detection results on testing set without noise (a) successfully verified code (b) unsuccessfully verified code with an undetected coded yarn.

sharply with the increasing ZMG noise level. Though the mAP for 0.1 variance ZMG noise testing set is 88.6%, the AP for coded yarn class “5” is 54.6%. Figure 9 demonstrates examples of detection results on testing set without noise. If one coded yarn cannot be detected or incorrectly detected in a specific tag, this tag will be classified to “unsuccessfully verified” after the checksum algorithm. Thus, the 0.1 variance ZMG noise testing set has a lower recognition success rate (66.7%)—6 out of 18 codes are unsuccessfully verified. Nevertheless, for the testing sets with different level of S&P noises, the recognition success rate remains the same (83.3%).

One important issue should be taken into consideration while applying deep learning-based algorithms, which is the computational cost. The training process of our tag recognition algorithm takes around 3 hours under the computational condition described

in Section 4. The computation time for each image in the testing set is around 0.5 seconds. It is essential to note that, once the algorithm is properly trained, it can be directly applied for tag recognition without updating parameters within the network. Therefore, the computational cost is acceptable even for industrial level application.

7. CONCLUSIONS AND FUTURE PERSPECTIVES

In the past decades, the implementation of traceability in manufacturing industry has gained more and more attention and provided solutions to several problems, such as counterfeit, product recall,

and environmental pollutions. Considering the limitations of conventional traceability tags (RFIDs and barcodes), textile-based tags is predicted to play a stronger role. Moreover, the booming of deep learning-based algorithms provided new paradigms for the coded yarn-based fabric tag recognition task. In this paper, new designs of the woven traceability tags with 3-ply coded yarns are proposed to increase tag diversity, and checksum algorithm is integrated for tag verification. A deep learning object detection algorithm—Faster R-CNN has been successfully introduced to provide a robust end-to-end solution for coded yarn recognition for the first time. The mAP for the coded yarn-based fabric tags detection reached 96.3% (without noise). The S&P noise has an insignificant effect on the mAP and code valid rate, whereas the ZMG noise has more impact on the mAP and code valid rate with the increase of ZMG noise variance. The checksum verification results demonstrated that 17 out of the total 18 testing images were detected correctly—94.4% recognition success rate (without noise), which was consistent with manual confirmation results. Therefore, it is promising to apply checksum to confirming whether the detection results are correct.

In future work, the potential feasibility for industrial application will be explored. For example, coded tag detection framework can be developed systematically. A pipeline including image capturing, code recognition, verification, and information exchange can provide a novel alternative in the real production scenario. The deep learning-based algorithm for the tag recognition can be improved accordingly for real-time detection. Moreover, we can introduce more possibility on the designing of coded yarns. The AP results of each yarn classes in this paper can also provide guidelines for the designing strategy, as the class “4” and “5” have relatively poor performance. It is reasonable to expect an increase in the AP with relatively low yarn twists. The 3-ply twisting yarn is proved an efficient way for yarn coding, two different colored yarns are utilized to offer contrast and “Z” twist is used in this paper for all the coded yarns. More color combination and different twist direction can provide more options for yarn coding. Further, the implementation of the coded yarns enables the traceability for woven fabrics in this paper, and can be expended to any kind of fabrics (e.g., knitted, nonwoven) with the help of embroidery technology, which opens up a new prospect. We must mention that the focuses of this paper is on the textile industry by selecting textile fabrics for demonstration, however, the use of deep learning as a tool makes our proposed concept and methodology highly adaptable to many other manufacturing sectors.

ACKNOWLEDGMENTS

The authors would like to thank Mr. Nicolas Dumont and Mr. Frédéric Veyet for their assistance with the preparation (spinning, weaving) of coded-yarn based traceability tags.

CONFLICT OF INTEREST

The authors declare no conflict of interest.

AUTHORS' CONTRIBUTIONS

K.W. conceived the research idea, designed the traceability tags, developed the tag recognition system, and wrote the manuscript; V.K. contributed in the designing and preparation of traceability

tags; X.Z. and L.K. contributed to the analysis of results; V.K. X.Z. L.K. X.T. and Y.C. contributed in refinement of the manuscript. All authors read and approved the manuscript.

FUNDING

This work was supported by Xuzhou Silk Fibre Science & Technology Co., Ltd.

REFERENCES

- [1] International Organization for Standardization, ISO 22005: traceability in the feed and food chain — general principles and basic requirements for system design and implementation, 2007. <https://www.iso.org/standard/36297.html>
- [2] C.J. Smith, Ensuring Supply Chain Security: The Role of Anti-Counterfeiting Technologies, 2016. http://www.unicri.it/topics/counterfeiting/anticounterfeiting_technologies/Ensuring_supply_chain_security_report.pdf
- [3] S. Tamayo, T. Monterio, N. Sauer, Deliveries optimization by exploiting production traceability information, *Eng. Appl. Artif. Intell.* 22 (2009), 557–568.
- [4] A. Abercrombie, E. Alexiev, D. Bral, C. Manager, B. Silvia Bollani, Informal Product Traceability Expert Group, final report, 2013. https://ec.europa.eu/info/sites/info/files/final-report_product-traceability-expert-group_2013_en.pdf
- [5] Rapanui Clothing, Traceability for clothing, 2016. <https://rapanui clothing.com/traceability-clothing/>
- [6] B. Xin, J. Hu, G. Baciú, X. Yu, Development of weave code technology for textile products, *Fibres Text. East. Eur.* 85 (2011), 33–35. <http://www.fibtex.lodz.pl/article497.html>
- [7] V. Kumar, L. Koehl, X. Zeng, A fully yarn integrated tag for tracking the international textile supply chain, *J. Manuf. Syst.* 40 (2016), 76–86.
- [8] V. Kumar, L. Koehl, X. Zeng, D. Ekwall, Coded yarn based tag for tracking textile supply chain, *J. Manuf. Syst.* 42 (2017), 124–139.
- [9] T.K. Agrawal, L. Koehl, C. Campagne, A secured tag for implementation of traceability in textile and clothing supply chain, *Int. J. Adv. Manuf. Technol.* 99 (2018), 2563–2577.
- [10] V. Kumar, C. Hallqvist, D. Ekwall, Developing a framework for traceability implementation in the textile supply chain, *Systems.* 5 (2017), 33.
- [11] X. Yao, J. Zhou, Y. Lin, Y. Li, H. Yu, Y. Liu, Smart manufacturing based on cyber-physical systems and beyond, *J. Intell. Manuf.* 24 (2017), 1–13.
- [12] V. Kumar, Exploring fully integrated textile tags and information systems for implementing traceability in textile supply chains, 2017. <http://ori.univ-lille1.fr/notice/view/univ-lille1-ori-454906>
- [13] T.C.T. Chen, Cloud intelligence in manufacturing, *J. Intell. Manuf.* 28 (2017), 1057–1059.
- [14] S. Corbellini, F. Ferraris, M. Parvis, A cryptographic system for brand authentication and material traceability in the textile industry, in 2006 IEEE Instrumentation and Measurement Technology Conference Proceedings, IEEE, Sorrento, 2006, pp. 1331–1335.
- [15] K. Kuusk, O. Tomico, G. Langereis, Crafting smart textiles – a meaningful way towards societal sustainability in the fashion field? *Nord. Text. J.* 1 (2012), 6–15. <http://hb.diva-portal.org/smash/get/diva2:869634/FULLTEXT01.pdf>
- [16] C. Michener, RFID on the factory floor - Protect the brand by integrating RFID at point of manufacture - RFID arena, 2019.

- <https://invengo.com/2015/protect-the-brand-by-integrating-rfid-at-point-of-manufacture/>
- [17] S. Kubler, W. Derigent, A. Thomas, E. Rondeau, Embedding data on “communicating materials” from context-sensitive information analysis, *J. Intell. Manuf.* 25 (2014), 1053–1064.
- [18] A.R. Köhler, L.M. Hilty, C. Bakker, Prospective impacts of electronic textiles on recycling and disposal, *J. Ind. Ecol.* 15 (2011), 496–511.
- [19] J. Kirtland, Identification Numbers and Check Digit Schemes, Mathematical Association of America, America, 2001. <https://books.google.fr/books?hl=en&lr=&id=Z8eka35WUb8C&oi=fnd&pg=PR9&dq=check+digit&ots=OZcN6li0Hz&sig=kiHACWocUU94wC6EJTKDtdeceKo#v=onepage&q=check+digit&f=false>
- [20] G. Csurka, M. Humenberger, From handcrafted to deep local invariant features, *ArXiv e-Prints*, America 2018. <https://arxiv.org/abs/1807.10254>
- [21] D.G. Lowe, Distinctive image features from scale-invariant keypoints, *Int. J. Comput. Vis.* 60 (2004), 91–110.
- [22] T.M. Mitchell, *Machine Learning*, McGraw-Hill Science/Engineering/Math, America, 1997. ISBN 0070428077.
- [23] G. Anthes, Deep learning comes of age, *Commun. ACM.* 56 (2013), 13.
- [24] K. He, X. Zhang, S. Ren, J. Sun, Deep residual learning for image recognition, *ArXiv e-Prints*, 2015.
- [25] J. Deng, W. Dong, R. Socher, L.-J. Li, K. Li, F.-F. Li, ImageNet: a large-scale hierarchical image database, in *IEEE Conference on Computer Vision and Pattern Recognition*, Miami, 2009, pp. 248–255.
- [26] M. Fu, P. Xu, X. Li, Q. Liu, M. Ye, C. Zhu, Fast crowd density estimation with convolutional neural networks, *Eng. Appl. Artif. Intell.* 43 (2015), 81–88.
- [27] L.H.S. Vogado, R.M.S. Veras, F.H.D. Araujo, R.R.V. Silva, R. T. Aires, Leukemia diagnosis in blood slides using transfer learning in CNNs and SVM for classification, *Eng. Appl. Artif. Intell.* 72 (2018), 415–422.
- [28] K. Ko, K. Sim, Deep convolutional framework for abnormal behavior detection in a smart surveillance system, *Eng. Appl. Artif. Intell.* 67 (2018), 226–234.
- [29] GS1, EAN/UPC barcodes, 2019. <https://www.gs1.org/standards/barcodes/ean-upc>
- [30] N. Dalal, B. Triggs, O. Gradients, D. Cordelia, N. Dalal, B. Triggs, Histograms of oriented gradients for human detection, in *IEEE Computer Society Conference on Computer Vision and Pattern Recognition*, CVPR 2005, San Diego, 2005, pp. 886–893.
- [31] B. Herbert, T. Tuytelaars, L. V. Gool, SURF: speeded up robust features, *European Conference on Computer Vision*, Springer, Berlin, Heidelberg, 2006.
- [32] A. Krizhevsky, I. Sutskever, G.E. Hinton, ImageNet classification with deep convolutional neural networks, in *Advances in Neural Information Processing Systems*, Lake Tahoe, 2012, pp. 1–9.
- [33] M.D. Zeiler, R. Fergus, Visualizing and understanding convolutional networks, in: D. Fleet, T. Pajdla, B. Schiele, T. Tuytelaars (Eds.), *Lecture Notes Computer Science*, vol. 8689, Springer, Switzerland, 2014, pp. 818–833.
- [34] S. Ren, K. He, R. Girshick, J. Sun, Faster R-CNN: towards real-time object detection with region proposal networks, *IEEE Trans. Pattern Anal. Mach. Intell.* 39 (2017), 1137–1149.
- [35] R. Girshick, J. Donahue, T. Darrell, U.C. Berkeley, J. Malik, Rich feature hierarchies for accurate object detection and semantic segmentation, in *The IEEE Conference on Computer Vision and Pattern Recognition*, Columbus, 2014, pp. 580–587.
- [36] R. Girshick, Fast R-CNN, in *The IEEE International Conference on Computer Vision*, Santiago, 2015, pp. 1440–1448.
- [37] C. Tan, F. Sun, T. Kong, W. Zhang, C. Yang, C. Liu, A survey on deep transfer learning, in *International Conference on Artificial Neural Networks*, Rhodes, Greece, 2018, pp. 270–279.
- [38] S.J. Pan, Q. Yang, A survey on transfer learning, *IEEE Trans. Knowledge Data Eng.* 22 (2010), 1345–1359.
- [39] J. Yosinski, J. Clune, Y. Bengio, H. Lipson, How transferable are features in deep neural networks? in *Advances in Neural Information Processing Systems*, Montréal, Canada, 2014, pp. 3320–3328. <https://arxiv.org/abs/1411.1792>
- [40] M. Everingham, L. Van Gool, C.K.I. Williams, J. Winn, A. Zisserman, The pascal visual object classes (VOC) challenge, *Int. J. Comput. Vis.* 88 (2010), 303–338.
- [41] J. Hosang, R. Benenson, B. Schiele, Learning non-maximum suppression, in *Proceeding - 30th IEEE Conference on Computer Vision and Pattern Recognition*, CVPR 2017, Honolulu, 2017, pp. 6469–6477.
- [42] A. Paszke, S. Gross, S. Chintala, G. Chanan, E. Yang, Z. DeVito, Z. Lin, A. Desmaison, L. Antiga, A. Lerer, Automatic differentiation in PyTorch, in *Advances in Neural Information Processing Systems*, Long Beach, CA, 2017, pp. 1–4. <https://openreview.net/forum?id=BJJsrmfCZ>
- [43] J. Yang, J. Lu, B. Dhruv, P. Devi, A faster pytorch implementation of faster R-CNN, 2017. <https://github.com/jwyang/faster-rcnn.pytorch>