Intersecting Machining Feature Localization and Recognition via Single Shot Multibox Detector

Peizhi Shi, Qunfen Qi, Yuchu Qin, Paul J. Scott, and Xiangqian Jiang

***Abstract*—In Industrie 4.0, machines are expected to become autonomous, self-aware and self-correcting. One important step in the area of manufacturing is feature recognition that aims to detect all the machining features from a 3-D model. In this research area, recognizing and locating a wide variety of highly intersecting features are extremely challenging as the topology information of features is substantially damaged because of the feature intersection. Motivated by the single shot multibox detector (SSD), this article presents a novel deep learning approach named SsdNet to tackle the machining feature localization and recognition problem. The typical SSD is designed for 2-D image objection detection rather than 3-D feature recognition. Therefore, the network architecture and output of SSD are modified to fulfil the purpose of this research. In addition, some advanced techniques are also utilized to further enhance the recognition performance. Experimental results on the benchmark dataset confirm that the proposed method achieves the state-of-the-art feature recognition performance (95.20% F-score), localization performance (90.62% F-score), and recognition efficiency (243.85 ms per model).**

***Index Terms*—Deep learning, feature recognition, Industrie 4.0, 3-D feature localization, single shot multibox detector (SSD).**

I. Introduction

IN THE realm of manufacturing, every product starts with a (or a set of) computer-aided design (CAD) model (or models). As we are now marching toward a new era of smart manufacturing (or so called Industrie 4.0), machines are expected to become autonomous, self-aware and self-correcting. One of the essential steps toward such advance, is the ability of a machine to "understand" a given CAD model, that is, recognize any machining features of the model. This is called feature recognition.

Manuscript received May 12, 2020; revised September 9, 2020; accepted October 7, 2020. Date of publication October 13, 2020; date of current version February 22, 2021. This work was supported in part by the EPSRC UKRI Innovation Fellowship (Ref. EP/S001328/1), in part by EPSRC Future Advanced Metrology Hub (Ref. EP/P006930/1), and in part by EPSRC Fellowship in Manufacturing (Ref. EP/R024162/1). Paper no. TII-20-2436. *(Corresponding author: Qunfen Qi.)*

The authors are with the EPSRC Future Advanced Metrology Hub, School of Computing and Engineering, University of Huddersfield, Huddersfield HD1 3DH, U.K. (e-mail: p.shi@hud.ac.uk; q.qi@hud.ac.uk; y.qin@hud.ac.uk; p.j.scott@hud.ac.uk; x.jiang@hud.ac.uk).

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

Digital Object Identifier 10.1109/TII.2020.3030620

Machining feature recognition has become an active research topic since 1980s, where a large number of methods have been proposed. Most methods were implemented based on manually designed rules. In these rule-based approaches, recognizing a wide variety of highly intersecting features remains a somewhat challenging task [1]–[4] as in-depth knowledge about different features and feature combinations is required. In recent years, machine learning techniques have been widely utilized in the area of smart manufacturing (e.g., machine fault diagnosis [5], predictive maintenance [6], 3-D object acquisition [7], retrieval [8], and recognition [9] for manufacturing automation), as well as in other research areas (e.g., image retrieval [10], medical volume segmentation [11]). These techniques enable intelligent agents to automatically learn from data without being explicitly programmed. To this end, two novel approaches named FeatureNet [12] and MsvNet [13] that adopt machine learning techniques for intersecting feature recognition have been developed. The two approaches are general purpose method in which ad hoc rules are not required any more, such that they can recognize a wide variety of features without imposing the burdens on the recognizer designer. In the two approaches, unsupervised segmentation algorithms were utilized to divide intersecting features into separated features according to the features' shape information. Then, the deep learning methods were employed to recognize these segmented features one by one. However, it is rather difficult to accurately segmenting intersecting features according to the shape information in an unsupervised way, since the topology information of the features might be destroyed because of feature intersection. This will lead to a large amount of features in a CAD model be misrecognized or mislocated, as evident in the experiments carried out in [13].

Motivated by an effective yet efficient object detection algorithm named single shot multibox detector (SSD) [14], this article presents a novel method called SsdNet where feature segmentation and recognition are carried out together via supervised learning. The typical SSD is a deep neural network designed for 2-D image objection detection, which cannot be directly employed to recognize 3-D models. Therefore, this article modified the network architecture and outputs of SSD to tackle intersecting 3-D machining feature localization and recognition problem. This article further utilizes data augmentation (DA) and transfer learning (TL) to improve the training performances.

The main contribution in this article is an approach named SsdNet capable of yielding the state-of-the-art intersecting feature recognition performance, localization performance, and recognition efficiency. As a minor contribution, a comprehensive

evaluation of the SsdNet and other learning-based approaches is also conducted in this article.

The rest of this article is organized as follows. Section II reviews the existing intersecting feature recognition methods and identifies the research gaps in this research area. Section III presents a novel method called SsdNet that overcomes the limitations arising from the existing methods. Section IV fully examines the performance of the SsdNet, and compares the SsdNet to other intersecting feature localization and recognition approaches. Finally, Section V concludes this article.

II. Related Work

Recognition and localization are two critical tasks in intelligent system development [15], [16]. In the area of manufacturing, feature recognition refers to the task for predicting the correct number and type of features appeared in the given CAD model, whereas feature localization refers to the task of finding the precise locations of features in the CAD model. Machining feature localization and recognition can be carried out by either rule-based [2] or learning-based approaches [17]. The former imply that human developers utilize the knowledge and experience to design rules for localization and recognition, whereas the latter aim to create feature recognizers via machine learning techniques from human labeled data. As isolated feature recognition problem has already been perfectly solved, this section will have a particular focus on an overview of the intersecting feature localization and recognition approaches.

A promising rule-based intersecting machining feature recognition approach is the hint-based approach [18]. In this approach, an important concept named *hint*, which refers to the minimum indispensable parts of a feature, was presented. During the recognition, the hint-based system first achieves all potential hints from a 3-D model. Then, a geometric completion procedure, which includes a heuristic geometric reasoning and matching procedure, is defined and adopted to find features from a given CAD model according to the hint instances. Both hint-based and other existing rule-based methods (e.g., STEP-based [19], [20], volumetric decomposition [21], and graph-based approaches [22]–[24]) suffer from a number of limitations: first, in-depth knowledge about different features is required to design a reliable rule-based approach; second, designing heuristic rules becomes more challenging in intersecting feature recognition, as the topology of a feature is destroyed and most faces in the feature are lost. Therefore, the rule developer has to consider all combinations of features, and carefully check whether the proposed rules (e.g., hints, geometric completion procedures) are valid in all the situations; third, most rule-based approaches adopt matching or searching algorithms to identify the potential feature in the 3-D model (e.g., the geometric completion procedure in hint-based approaches), which is computationally expensive [2].

As noted, designing heuristic rules for intersecting feature localization and recognition is not an easy task. Some learning-based approaches have been applied to reduce the effort required for the rule developers. However, most of these approaches (e.g., [25]–[27]) can only tackle limited types of feature intersections, and/or focus on specific type of CAD representations. To tackle the abovementioned issues, Zhang *et al.* [12] presented a feature recognizer named FeatureNet, which can locate and recognize any types of intersecting features in a given CAD model. In this approach, an unsupervised segmentation algorithm, called watershed algorithm, was employed to divide intersecting features into separated features according to the features' shape information. A 3-D convolutional network was then utilized to recognize these segmented features one by one. In general, watershed algorithm can yield expected results when segmenting features with low overlap degree, but fails to separate highly intersecting features as the shape information of most features is lost because of the feature intersection. To solve the issues arising from the FeatureNet, Shi *et al.* [13] proposed a novel intersecting feature recognition approach named MsvNet. In this approach, a 3-D model with intersecting features was first segmented into separated ones via another unsupervised learning algorithm named selective search algorithm according to the 2-D shape information of the features. Then, these segmented features were passed through a novel view-based 2-D convolutional neural network (CNN) for further recognition. Unlike the watershed algorithm that only produces one set of segmentation results based on one 3-D model, the selective search algorithm can enumerate most potential features in a 3-D model. Therefore, more intersecting features are likely to be found by the selective search algorithm, which leads to a better localization and recognition performance than the FeatureNet. Both the FeatureNet and MsvNet suffer from the following limitations: first, due to the nature of unsupervised segmentation algorithms involved in these methods, a large number of highly intersecting features could still be misrecognised or mislocated as the topology information of these features is substantially damaged because of the feature intersection. Therefore, unsatisfactory localization and recognition results could be produced, which is also illustrated in the experiments; second, the FeatureNet and the MsvNet can be regarded as two-stage methods in which feature segmentation and recognition are conducted separately. Therefore, segmented features need to be passed through the neural networks multiple times, which will slow down the whole recognition process.

III. Framework

This section first discusses the relevant issues raised in the existing approaches, which motivate the proposed method, and makes an overview on the proposed approach to intersecting feature localization and recognition. Then, the neural network construction process and the final feature localization and recognition process are illustrated in details.

A. Overview

As discussed in Section II, existing learning-based methods (MsvNet [13] and FeatureNet [12]) suffer from a number of limitations, which motivate the research conducted in this article. Therefore, the main research problem that this article aims to tackle is: how to locate and recognize highly intersecting machining features from a CAD model efficiently. To solve this

research problem, several advanced methods are explored in this article.

Both the MsvNet and FeatureNet are two-stage methods with the abovementioned limitations. A one-stage method that conducts feature segmentation and recognition together via supervised learning seems to be a proper solution to the abovementioned issues. In a supervised algorithm, different kinds of intersecting features can be seen at the training stage rather than the test stage, which allows for producing much better segmentation and recognition performances. In a one-stage algorithm, feature segmentation and recognition are carried out together, which could speed up the recognition process.

As evident in the experiments conducted in [13], segmenting intersecting features in 3-D space is rather arduous. Experimental results also demonstrated that it was relatively easy to locate and recognize 3-D intersection features from 2-D view images [13]. To this end, a one-stage supervised feature segmentation and recognition algorithm based on 2-D view images is an ideal solution to the research problem. In other words, the proposed deep neural network takes a view image as input, and predicts the types and 3-D locations of all features appeared in this view direction. Finally, the 3-D bounding boxes achieved from different view directions are combined together to form the final results. In summary, the SsdNet consists of two parts: one-stage supervised feature localization and recognition (to predict the types and locations of features appeared in different view directions), and result fusion (to form the final prediction results). The machine learning techniques employed in each part are shown in the next two sections, respectively.

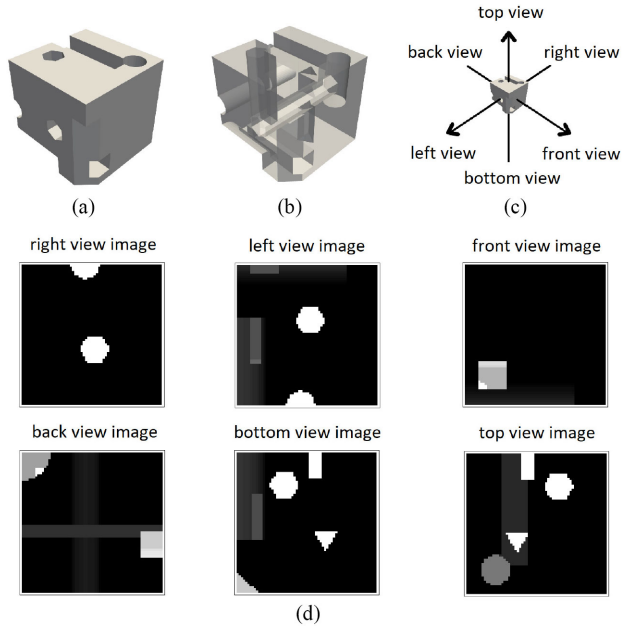


Fig. 1. 2-D images taken from different directions. (a) Original model. (b) Transparent model. (c) View labels in six directions. (d) Images with view labels. Each pixel value of images in (d) represents the maximal depth of all features appeared at this position.

B. Network Construction

The main purpose of this section is to construct a deep neural network that maps a 2-D view image to 3-D locations of all features appeared in this view direction. To attain this goal, SSD [14], an effective yet efficient one-stage object detection algorithm, is adopted in this article, as it is capable of identifying objects appeared in an image effectively. The original SSD is designed for image object detection, where the output of the algorithm is a set of 2-D bounding boxes. It means that it cannot be applied to the research problem directly since the output in feature localization should be 3-D locations of the features rather than 2-D bounding boxes. Therefore, this article adjusts the output of the original SSD to tackle the 3-D feature localization and recognition problem. In addition to the output, the architecture of the SSD is also modified to make the training and recognition processes more efficient.

This section first discusses how to prepare the data for training. Then, the novel network architecture designed based on the research problem and its training process are presented in details.

1) Data Preparation: As the SSD-based approach is supervised learning, 3-D models with intersecting features and their corresponding labels are required at the training stage. In practice, however, it may be easier to get 3-D models with single features than models with intersecting features. To tackle this problem, this approach synthesises multifeature models by combining single feature models together. In this article, it is assumed that a dataset which consists of different types of 3-D voxelized single feature models is available. Before training, a number of 3-D models with single features (e.g., two–ten models) are randomly selected from this dataset, and combined together via the boolean operation to form a multifeature model. The types and bounding boxes of all the features in this newly constructed model are also recorded, and denoted as the label of the 3-D model. Then, a large number of 3-D multifeature models with labels can be constructed effectively.

As discussed in Section II and [13], feature segmentation in a 3-D space is much more challenging than in a 2-D space. Therefore, a 3-D model with intersecting features is converted into a number of view images in this article. These images are employed as inputs of the proposed network as it will allow for training an effective and efficient network for feature segmentation and recognition easily. Suppose that the dimension of a voxelized 3-D model is $d \times d \times d$. This approach scans this model from six directions, and takes six $d \times d$ images from this model accordingly. In each image, the value of each pixel represents the maximal depth of all features appeared at this position, as exemplified in Fig. 1.

To improve the training performances, DA, a widely used technique in the area of machine learning, is adopted in this approach. This method is able to considerably increase the diversity of training samples. In this article, three DA strategies are employed: *random flipping*, *random resizing*, and *random combination*. In the first strategy, the 2-D image is horizontally or vertically flipped with a small probability. This strategy is capable of producing new training images that contain features with different locations. In the second strategy, constant padding is applied to the top, bottom, left, and right of the 2-D training

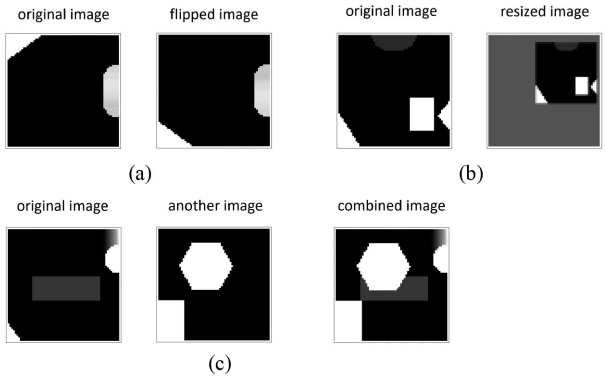


Fig. 2. DA strategies. (a) Random flipping. (b) Random resizing. (c) Random combination.

image. This padded image is finally resized to the original size. This strategy could produce new training images with smaller machining features. In the third strategy, the 2-D image is combined with another training image via the element-wise maximum operation. The labels (or bounding boxes) of two images are also concatenated together. This strategy allows for producing training images, which contain more machining features. The detailed DA process is illustrated in Fig. 2.

2) Network Architecture: The original SSD [14] is a feed-forward CNN, which maps a 2-D image to a set of 2-D bounding boxes and class scores of objects appeared in this image. It employed a high-resolution image ($3 \times 300 \times 300$) as input, which is computationally expensive. The output of the proposed framework is supposed to be a set of 3-D feature bounding boxes rather than 2-D boxes. Therefore, this article adjusts the architecture and output of the original SSD to fulfil the purpose of this article. The modified network takes a smaller 2-D view image as input, and predicts a set of 3-D bounding boxes and class scores of all features appeared in this view direction. Such a modification could produce a much better result in terms of recognition accuracy and efficiency.

As shown in Table I, the network contains two components: a *base net* and a *multibox net*. The former is employed to produce six activation maps with different sizes based on the input view image, whereas the latter is utilized to predict the types and locations of the 3-D machining features at multiple scales based on the six activation maps.

It is also observed from Table I that the network contains 25 convolutional layers in total (19 in the base net, 6 in the multibox net). Each convolutional layer consists of a fixed number of kernels, which are employed to apply some effects to the inputs of neurons. The kernel operation is conducted as

$$z(i,j,k) = \sum_{l,m,n} q(l, j+m-1, k+n-1)k(i,l,m,n) \quad (1)$$

where q is the input of the neuron, $z(i,j,k)$ is the output at a location (j,k) for the ith channel, and k is the kernel matrix. In addition to the convolutional layers, three $l2$ norm layers [28] are adopted to normalize the activation maps achieved from the earlier convolutional layers since the earlier activation

TABLE I
NETWORK ARCHITECTURE TABLE. THIS NETWORK CONSISTS OF A BASE NET AND A MULTIBOX NET. THE BASE NET IS UTILIZED TO CREATE SIX ACTIVATION MAPS, WHEREAS THE MULTIBOX NET IS EMPLOYED TO PRODUCE THE FINAL RESULTS BASED ON THE ACTIVATION MAPS

Input	Base Net	Activation Map	Multibox Net	Output
$3\times64\times64$	Conv 64 Conv 64 Max Pool Conv 128 Conv 128	$\rightarrow 128\times32\times32$	$\rightarrow$ Norm Conv $6\times$(c+5)	$\rightarrow 6\times$(c+5)$\times32\times32$
	$\downarrow$ Max Pool Conv 256 Conv 256 Conv 256	$\rightarrow 256\times16\times16$	$\rightarrow$ Norm Conv $6\times$(c+5)	$\rightarrow 6\times$(c+5)$\times16\times16$
	$\downarrow$ Max Pool Conv 512 Conv 512 Conv 512	$\rightarrow 512\times8\times8$	$\rightarrow$ Norm Conv $6\times$(c+5)	$\rightarrow 6\times$(c+5)$\times8\times8$
	$\downarrow$ Max Pool Conv 512 Conv 512 Conv 512 Max Pool Conv 1024 Conv 1024	$\rightarrow 1024\times4\times4$	$\rightarrow$ Conv $6\times$(c+5)	$\rightarrow 6\times$(c+5)$\times4\times4$
	$\downarrow$ Conv 256 Conv 512	$\rightarrow 512\times2\times2$	$\rightarrow$ Conv $6\times$(c+5)	$\rightarrow 6\times$(c+5)$\times2\times2$
	$\downarrow$ Conv 128 Conv 256	$\rightarrow 256\times1\times1$	$\rightarrow$ Conv $6\times$(c+5)	$\rightarrow 6\times$(c+5)$\times1\times1$

maps usually have larger values than the latter maps. After the $l2$ normalization, all activation maps will have a similar value range. The $l2$ norm operation is defined as

$$o(i,j,k) = \gamma_i \frac{z(i,j,k)}{\sqrt{\sum_i |z(i,j,k)|^2}} \quad (2)$$

where γ_i is a learnable scaling factor for the channel i.

For an activation map, six predefined reference bounding boxes are associated with each cell of the activation map in this approach [see the dotted line boxes in Fig. 3(a)]. It is observed from Fig. 3(a) that each reference box is centred at the cell of the activation map, and has fixed shape and size. The bounding box for a machining feature, however, is supposed to have arbitrary shape, size, and location [see the purple dashed line box in Fig. 3(b)]. To attain this goal, the multibox network predicts offsets of the machining feature bounding box relative to the predefined reference bounding box, and the confidence score for each type of machining feature [as shown in Fig. 3(b)]. Therefore, $6 \times m \times m$ reference bounding boxes can be predefined based on the activation map of size $m \times m$, and $6 \times m \times m$ machining feature bounding boxes can be constructed based on these predefined reference boxes. The dimension of the final predictions is $6 \times (c+5) \times m \times m$, where 6 refers to the number of machining feature bounding boxes per cell, c is the number of feature types and 5 refers to the five offset values

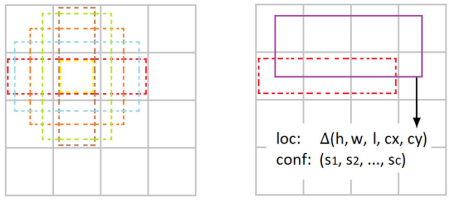


Fig. 3. Reference bounding boxes and machining feature bounding boxes in the activation map (adapted from [14]). (a) 4×4 activation map and the six corresponding reference bounding boxes (the dotted line boxes) in one cell. (b) Machining feature bounding box (the purple dashed line box) and its corresponding reference box (the red dotted line box). $\Delta(h, w, l, cx, cy)$ in (b) refer to the location offset values of the purple machining feature box relative to the red reference box. $(s_1, ..., s_c)$ refer to the confidence scores for all feature types.

[height h, width w, depth l, and centre coordinate (cx, cy)] of the machining feature bounding box relative to the predefined reference box [as illustrated in Fig. 3(b)].

3) Training: In the proposed deep neural network, it is essential to know which reference bounding box is responsible for predicting a certain ground truth bounding box. To tackle this issue, Liu *et al.* [14] presented a positive and negative matching strategy to find corresponding reference bounding boxes for a ground truth bounding box. In this approach, the ratio between the overlapped area over the joined area, called intersection over union (IoU) value [13], is utilized to measure the degree of overlap between two bounding boxes. A match is positive when IoU value between two boxes is greater than 0.5. The rest of matches are regarded as negative. It is obvious that there are more negative matches than positive ones, which makes the training data extremely imbalanced. Therefore, this approach only selects negative reference bounding boxes with top loss values and guarantees that the ratio between the selected positive ones and negative ones is 1:3, as suggested in [14].

Suppose that there is a matching between a ground truth bounding box t and reference box d. The width, height, depth, and centre coordinate of a box are denoted as w, h, l, and (cx, cy), respectively. The encoded offsets of the ground truth box t relative to the reference box d are defined as

$$\hat{t}^w = \log(t^w/d^w), \quad \hat{t}^h = \log(t^h/d^h), \quad \hat{t}^l = \log(t^l/d^l)$$
$$\hat{t}^{cx} = (t^{cx} - d^{cx})/d^w, \quad \hat{t}^{cy} = (t^{cy} - d^{cy})/d^h \tag{3}$$

where t^w, t^h, t^l, and (t^{cx}, t^{cy}) refer to the width, height, depth, and centre coordinate of the ground truth bounding box. d^w, d^h, d^l, and (d^{cx}, d^{cy}) refer to the width, height, depth, and centre coordinate of the reference bounding box as illustrated in Fig. 3. $\hat{t}^w$, $\hat{t}^h$, $\hat{t}^l$, $(\hat{t}^{cx}, \hat{t}^{cy})$ refer to the width, height, depth, and centre coordinate offset values of the ground truth box t relative to the reference box d.

In this approach, confidence loss (L_{conf}) and localization loss (L_{loc}) are employed to train the neural network. The former measures how confident the deep network is of making a class prediction, whereas the latter is the mismatch between the predicted box and ground truth box. The confidence loss is defined as

$$L_{\text{conf}}(x, s) = -\sum_{i \in \text{Pos}} x_{i,j}^k \log\left(\frac{e^{s_i^k}}{\sum_k e^{s_i^k}}\right) - \sum_{i \in \text{Neg}} \log\left(\frac{e^{s_i^0}}{\sum_k e^{s_i^k}}\right) \tag{4}$$

where $x_{i,j}^k$ is an indicator value, which equals to one when there is a match between the ith reference bounding box and the jth ground truth box for the feature type k. s_i^k refers to the predicted confidence score for the feature type k achieved from the ith reference box. e is the exponential constant approximately equal to 2.71828. The localization loss is defined as

$$L_{\text{loc}}(x, p, t) = \sum_{m \in \{cx, cy, w, h, l\}} \sum_{i \in \text{Pos}} x_{i,j}^k \text{Smooth}_{\text{L1}}(p_i^m - \hat{t}_j^m), \tag{5}$$

where p_i^m refers to the predicted localization offset based on the ith reference box, and $\text{Smooth}_{\text{L1}}$ is the Smooth L1 loss [29]. This loss function is selected since it is less sensitive to outliers than other loss functions, and is capable of preventing exploding gradients during training [29]. The overall loss for all matches is calculated as

$$L(x, s, p, t) = \frac{1}{N}(L_{\text{conf}}(x, s) + L_{\text{loc}}(x, p, t)) \tag{6}$$

where N refers to the number of matches.

At the beginning of the training stage, network parameter initialization is an important step since a better learning result could be achieved from a well-initialized neural network. To tackle this problem, TL, a popular method for knowledge transfer and parameter initialization, is adopted in this article. This technique is capable of employing the knowledge gained from one problem to solve another problem. In general, the network trained on a dataset of visual objects contains deep knowledge of object detection, and could be utilized to initialize the parameters (e.g., weights and biases) in the another network for object detection. Therefore, this article employs a pretrained SSD network on the pascal visual object classes (VOC) benchmark set [30] to initialize parameters in the proposed network since the VOC is a large set for object detection. The detailed process of utilizing TL is presented in Section IV-B. During the training, this approach employs the Adam optimizer to minimize the loss function $L(x, s, p, t)$, as this optimizer can converge to minimum faster than other optimizers.

C. Feature Localization and Recognition

As illustrated in the previous section, a neural network which maps a 2-D view image to a number of possible 3-D feature locations is constructed. Therefore, the next issue is how to utilize this network for feature localization and recognition based on a 3-D model rather than the 2-D images. To attain this goal, the six view images of the 3-D model are first passed through the network. Then, the outputs of the neural network are decoded [see Fig. 4(a) and (b)] as follows:

$$\bar{t}^w = d^w e^{p^w}, \quad \bar{t}^h = d^h e^{p^h}, \quad \bar{t}^l = d^l e^{p^l}$$
$$\bar{t}^{cx} = p^{cx} d^w + d^{cx}, \quad \bar{t}^{cy} = p^{cy} d^h + d^{cy} \tag{7}$$

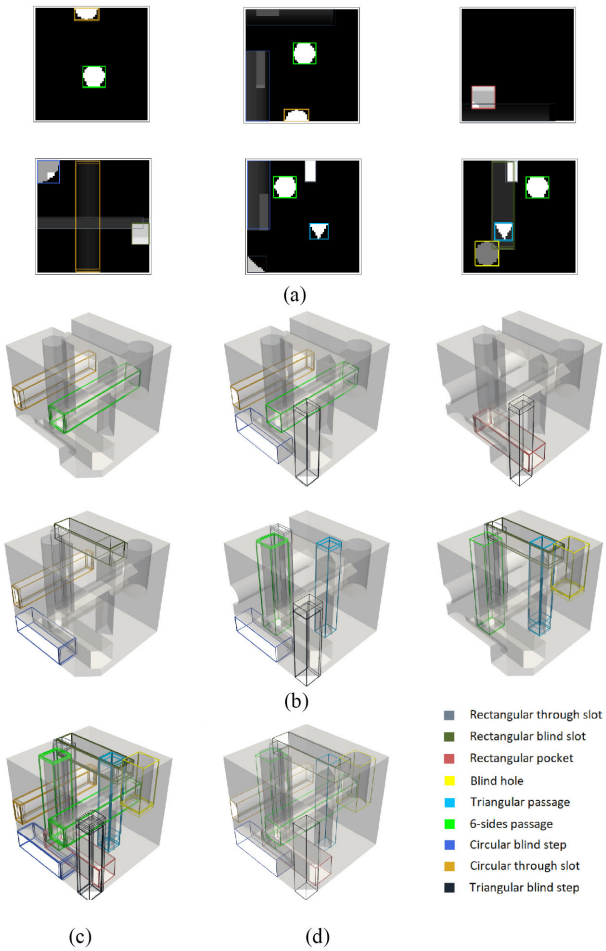


Fig. 4. Result fusion process. (a) Predicted features achieved from each direction in 2-D images (the depth information about each bounding box is not displayed). (b) Predicted features achieved from each direction in 3-D models. (c) Combined features in a 3-D space. (d) Final result.

where p^w, p^h, p^l, and (p^{cx}, p^{cy}) refer to the predicted width, height, depth, and centre coordinate offset values based on the reference box d. $\bar{t}^w$, $\bar{t}^h$, $\bar{t}^l$, and $(\bar{t}^{cx}, \bar{t}^{cy})$ refer to the decoded width, height, depth, and centre coordinate of predicted bounding box.

After decoding, six sets of machining features could be produced based on the six view images of the 3-D model [see Fig. 4(b)]. Then, the bounding boxes of these features are concatenated together, as illustrated in see Fig. 4(c). It is observed that there are several redundant features in the result achieved from the previous step since the neural network is designed to find all potential features from a given CAD model. To remove these features, soft nonmaximum suppression (Soft-NMS) [31], the state-of-the-art bounding box selection method, with maximum cut algorithm [32] is adopted. As suggested in [13], this method is capable of eliminating redundant features effectively.

The Soft-NMS [31] algorithm starts with a list of 3-D location bounding boxes $\mathcal{B} = \{b_1, ..., b_n\}$ and their corresponding localization-recognition scores $\mathcal{S} = \{s_1, ..., s_n\}$. In this algorithm, both the bounding boxes and corresponding score values are captured from the output layer of the SsdNet. Then, a greedy procedure is carried out to move a 3-D bounding box b_m with the highest score value from $\mathcal{B}$ to a new bounding box set $\mathcal{D}$, and reduce the score value of each bounding box b_i in $\mathcal{B}$ proportional to the IoU values between the b_m and b_i. This greedy procedure is terminated when all boxes in $\mathcal{B}$ are moved to $\mathcal{D}$. At the end, boxes in $\mathcal{D}$ with high score values are selected as final results via the max-cut algorithm [32]. It is observed from Fig. 4(d) that, this approach effectively removes redundant and wrongly recognized machining features.

IV. Experimental Results

Based on the framework presented in the previous section, this section first makes a comparison between the proposed approach and other learning-based approaches (the MsvNet [13] and FeatureNet [12]) in terms of intersecting machining feature localization and recognition. Then, the effects of different learning strategies in the SsdNet are further examined. The source code of the proposed framework as well as the experimental results is available online[1].

A. Benchmark Dataset

As shown in Section III and [12], [13], a single feature dataset is required to fully train the SsdNet, MsvNet, and FeatureNet. Therefore, the *benchmark single feature set* constructed in [12] is adopted in this experiment since it is a diverse set with 24 different types of machining features. In total, there are 24 000 3-D STL models in this set, 1000 for each type of features.

In addition to the single feature dataset, a multifeature dataset is also required to test the localization and recognition performances of different approaches. Therefore, the *benchmark multifeature set* presented in [13] is employed in this experiment for testing purpose since this set consists of 1000 STL models with highly intersecting features. Shi *et al.* [13] divided the dataset into ten different groups according to the intersecting degree of features.

All the methods in this comparative study require 3-D voxelized models for training and testing. Therefore, a toolbox named *binvox* is employed to convert 3-D STL models in two benchmark sets into 3-D $64 \times 64 \times 64$ grids as carried out in [12] and [13]. Therefore, each set contains models with shape $64 \times 64 \times 64$. To make a fair comparison, all the experiments are conducted under an identical optimal setting as suggested in [13]. The abovementioned networks are trained and validated on the benchmark single feature set [12], and tested on the benchmark multifeature set [13]. The single feature set is divided into training and validation sets (90%:10%). At the training phase, only 512 models per feature type (51.2%) are utilized to train the networks, the same as in [13]. All the models in the benchmark multifeature set are selected to form a test set. The information about the training, validation, and test sets is summarized in Table II.

[1][Online]. Available: https://github.com/PeizhiShi/SsdNet

TABLE II
DATASET DESCRIPTION

	Training Set	Validation Set	Test Set
Source	FeatureNet Set [12]	FeatureNet Set [12]	MsvNet Set [13]
Dimension	$64 \times 64 \times 64$	$64 \times 64 \times 64$	$64 \times 64 \times 64$
Property	single feature models	single feature models	multi-feature models
Feature Type	24 machining features	24 machining features	24 machining features
Set size	512 models per type	100 models per type	1000 models in total

TABLE III
F-SCORE FOR FEATURE RECOGNITION (%)

	Test data group [13]										
Method	all	1	2	3	4	5	6	7	8	9	10
SsdNet	95.20	99.33	98.82	98.02	97.38	97.02	95.77	95.53	94.08	93.26	90.91
MsvNet	76.24	95.93	87.95	83.87	78.22	76.33	76.83	77.49	75.14	70.90	66.77
FeatureNet	57.45	92.83	81.01	64.35	59.09	60.13	57.01	54.78	53.80	50.18	46.49

B. Experimental Settings

In the SsdNet, 2-D images instead of 3-D models are required for training the deep network. Therefore, 2.8 M training images and 1 K validation images are created based on the 3-D models in the training and validation sets by following the procedures described in Section III-B1. As stated in Section III-B2, the proposed network consists of a base net and a multibox net. TL adopts a pretrained SSD network on the VOC dataset to initialize parameters in the base net since the structures of the base nets in the original SSD network [14] and the proposed SsdNet are identical. The bias in each neuron of the multibox net is set to zero, whereas each weight in the multibox net is set to a small random number. Other technologies in TL (e.g., weight freezing) are not adopted in this article. The batch size is set as 16, whereas the number of learning epochs is set to 4 (700 000 training steps in total). The probability of applying each DA strategy to the training images is 50%. The learning rate is initially set as 10^{-4} and then set as 10^{-5} in the third epoch. This simple learning rate decay scheme for Adam is utilized since it is able to yield better learning results [33]. The values of the aforementioned hyperparameters are determined according to the validation loss. In the MsvNet and FeatureNet, the values of all the hyperparameters are identical to those in [13]. It is worth noting that the intersecting feature segmentation part in the FeatureNet is only a reimplemented version provided by Shi *et al.* [13] where the watershed algorithm with a default configuration is utilized. An Intel i9-9900X PC with a 128 GB memory and NVIDIA RTX 2080ti GPU is employed to carry out the experiments reported in the following sections.

For a machining feature detector, it is important to measure its ability to locate and recognize the appeared features. As suggested in [34], *F-score* is adopted in this comparative study as this metric is suitable for multiobject classification and detection problem [35]. The F-score is the weighted average of *precision* and *recall*. The precision is the average fraction of correctly recognized/located features (true positive) among all the recognized/located features, the recall is the average fraction of correctly recognized/located features (true positive) among the total appeared features.

C. Recognition Performance

This section focuses on examining the recognition performances of different approaches. Therefore, F-score is employed as evaluation metric, where the true positive value tp_i for a 3-D model is calculated as

$$\mathrm{tp}_i = \min(\mathrm{pred}_i, \mathrm{gt}_i) \tag{8}$$

as implemented in [13]. tp_i refers to the true positive value which is the number of correctly recognized type i feature in a 3-D model, pred_i is the number of predicted type i feature in this model, and gt_i is the actual number of the type i feature appeared in this model. For instance, a 3-D model contains five holes ($\mathrm{gt}_{\mathrm{hole}} = 5$) and two pockets ($\mathrm{gt}_{\mathrm{pocket}} = 2$). The feature recognizer, however, reports that there are four holes ($\mathrm{pred}_{\mathrm{hole}} = 4$) and three pockets ($\mathrm{pred}_{\mathrm{pocket}} = 3$) appeared in this 3-D model. Therefore, the number of correctly recognized holes and pockets in this model should be four [$\mathrm{tp}_{\mathrm{hole}} = \min(\mathrm{pred}_{\mathrm{hole}}, \mathrm{gt}_{\mathrm{hole}}) = \min(5, 4)$] and two [$\mathrm{tp}_{\mathrm{pocket}} = \min(\mathrm{pred}_{\mathrm{pocket}}, \mathrm{gt}_{\mathrm{pocket}}) = \min(2, 3)$], respectively. Such a calculation only focuses on the evaluation of recognition performance without considering whether the predicted features are located correctly.

Table III shows the F-score for feature recognition on different data groups. As illustrated in the table, the SsdNet achieves the highest recognition F-score for all groups, which means that the proposed method is capable of producing more correct predictions than incorrect ones, and also finding more features from CAD models. As discussed in Section II, the MsvNet and FeatureNet were proposed based on unsupervised segmentation algorithms, which are not very suitable for 3-D models with highly intersecting features since the shape information of most features are damaged because of feature intersection. Therefore, the SsdNet could produce much better results as supervised segmentation algorithm is utilized.

D. Localization Performance

While the recognition performances of different approaches were examined in the previous section, this section further evaluates whether these approaches can accurately find the locations of the features from the CAD model. In this experiment, the F-score metric is also employed, but the way of calculating the true positive (tp_i) is different. As suggested in [34], a detection is considered as true positive only when the IoU value between the ground truth and predicted boxes is greater than 0.5. If multiple prediction boxes match a same ground-truth box, this metric only keeps the box with the top prediction score. For instance, there are five holes in a 3-D model ($\mathrm{gt}_{\mathrm{hole}} = 5$). The system finds four holes from this model ($\mathrm{pred}_{\mathrm{hole}} = 4$). Among the four holes, only one hole is located precisely. Therefore, the number of correctly recognized yet located holes in this model should be one instead of four ($\mathrm{tp}_{\mathrm{hole}} = 1$). Such a calculation allows for evaluating whether the predicted features are located correctly.

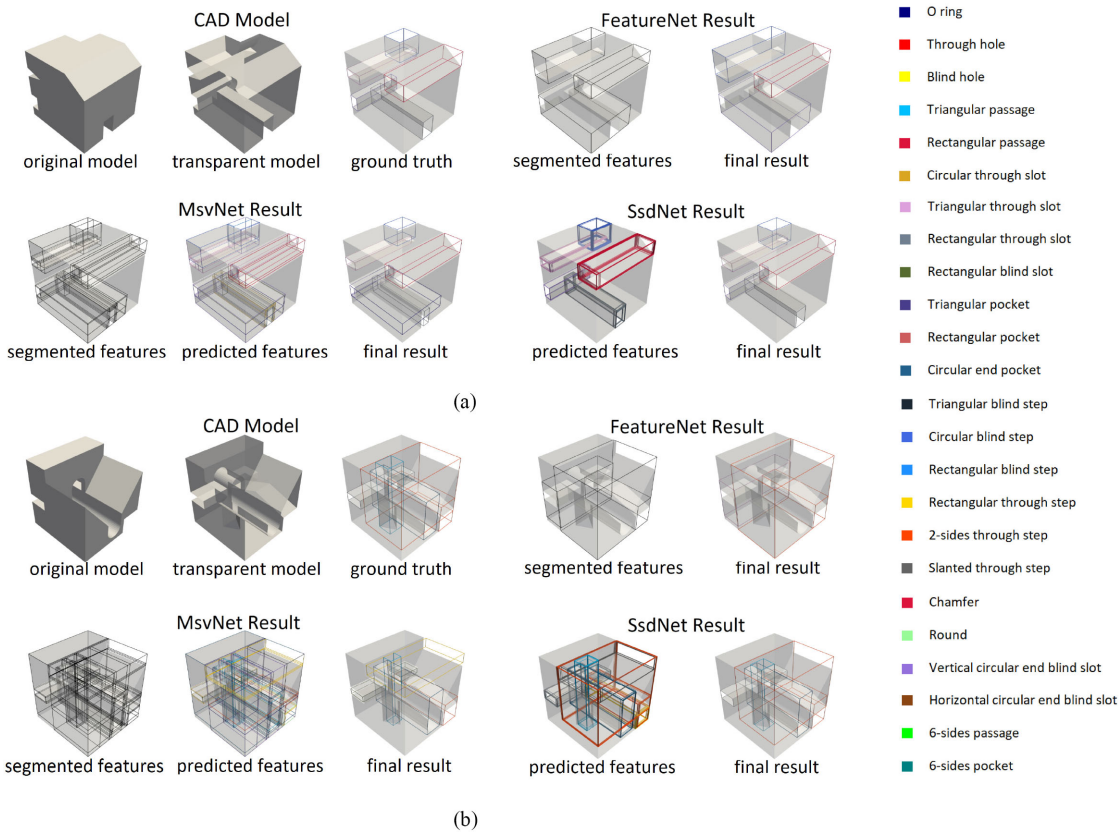


Fig. 5. Two 3-D models with intersecting features and their predicted bounding boxes yielded by the FeatureNet, MsvNet, and SsdNet. The original CAD model in (a) contains five features with medium degree of overlap, whereas the original model in (b) consists of five features with high degree of overlap. The intermediate and final results (e.g., results achieved from feature segmentation, recognition, and selection) yielded by three approaches are presented.

TABLE IV
F-SCORE FOR FEATURE LOCALIZATION (%)

	Test data group [13]										
Method	all	1	2	3	4	5	6	7	8	9	10
SsdNet	90.62	96.73	96.22	93.53	92.74	92.59	91.57	90.90	89.24	88.19	85.09
MsvNet	58.26	85.36	77.20	66.22	60.72	60.99	59.51	59.76	56.66	50.15	44.33
FeatureNet	38.37	88.01	70.04	51.48	43.73	41.75	40.75	34.85	31.72	24.84	19.26

Table IV illustrates the F-score for feature localization on different data groups. It is evident from the table that the SsdNet produces the highest F-score for all groups, especially when recognizing models with highly intersecting features (e.g., models in group 7–10). To fully examine the reason for this, Fig. 5 illustrates two 3-D models with intersecting features and their predicted bounding boxes achieved from different approaches. The original CAD model in Fig. 5(a) consists of five features: a rectangular through slot, a vertical circular end blind slot, a triangular through slot, a chamfer, and a circular blind step. Among these features, the rectangular through slot and the vertical circular end blind slot are overlapped together; the triangular through slot and the circular blind step are also intersecting features. In the FeatureNet [12], an unsupervised learning algorithm named watershed algorithm was first employed to segment features according to the their 3-D shape information. From the segmentation result achieved from the FeatureNet in Fig. 5(a), it is evident that this algorithm fails to segment these intersecting features appeared in the given CAD model. The MsvNet [13] employed another unsupervised learning algorithm named selective search algorithm to segment the features. Unlike the watershed algorithm that only produces one set of segmentation results based on one 3-D model, the selective search algorithm aims to enumerate all (or most) possible features in a given 3-D CAD model [see the segmentation result achieved from the MsvNet in Fig. 5(a)]. Therefore, most intersecting features are very likely to be found by the selective search algorithm, which could lead to a better localization performance than the FeatureNet. Due to the nature of unsupervised learning, however, six instead of five features are detected by the MsvNet [see the final result achieved from the MsvNet in Fig. 5(a)]. For 3-D models with highly intersecting features, the topology of each feature may be destroyed. In these situations, using unsupervised segmentation algorithms for feature segmentation and localization is particularly arduous. The SsdNet is a one-stage method which directly segments, locates and recognizes the intersecting features via supervised learning algorithm. From the final result achieved from the SsdNet in Fig. 5(a), different types of intersecting features can be identified correctly since supervised segmentation algorithm is employed. The CAD model in Fig. 5(b) also contains five features, while

TABLE V
COMPARISON TABLE

Method	Average Time (ms)						Average Number	
	(1) All	(2) Pre-processing	(3) Segmentation	(4) Recognition	(5) Post-processing	(6) Selection	(7) Segmented Features	(8) Forward Passes
SsdNet	243.85	145.04	-	9.49	59.77	18.19	65.55	6
MsvNet	724.18	131.64	373.32	145.84	51.6	6.67	24.46	24.46
FeatureNet	381.65	-	314.53	48.20	11.06	-	6.12	6.12

four of them are overlapped together. It is observed that the FeatureNet fails to segment these four intersecting features correctly, which leads to the wrong localization result [see the final result achieved from the FeatureNet in Fig. 5(b)]. From this figure, it is evident that the MsvNet is capable of detecting two features correctly, whereas the SsdNet could locate and recognize all these highly intersecting features easily even when the shape information of these features is substantially damaged due to the feature intersection.

E. Efficiency

This section further compares the proposed method to others in terms of the efficiency since the runtime performance of a feature recognition system is critical to computer-aided manufacturing. In this experiment, the following evaluation metrics are utilized.

1) The average time taken by different methods in recognizing a 3-D model in the test set.
2) The average time taken in data preprocessing (e.g., converting a 3-D model into a set of 2-D images).
3) The average time taken in feature segmentation.
4) The average time taken in feature recognition (e.g., forward pass).
5) The average time taken in the postprocessing (e.g., converting the outputs of the network into a set of 3-D bounding boxes).
6) The average time taken in feature selection.
7) The average number of segmented features.
8) The average number of forward passes.

For a fair comparison, all the experiments are conducted on an Intel i9-9900X PC with a 128 GB memory and NVIDIA RTX 2080ti GPU.

From the Table V, it is observed that the SsdNet is the most efficient intersecting machining feature localization and recognition method (243.85 ms per model). As stated in the previous sections, the SsdNet and MsvNet employs 2-D images rather than 3-D models as inputs. Therefore, these two approaches take similar constant times for preprocessing the input data (145.04 and 131.64 ms per model, respectively). As discussed in Sections II and III, the MsvNet and FeatureNet are two-stage methods in which feature segmentation and recognition are conducted separately. In these approaches, machining features are first separated via unsupervised algorithms. Then, the neural networks need to recognize these segmented features one by one, which is time-consuming. The SsdNet, however, is an one-stage method where feature segmentation and recognition are carried out together. It predicts the feature types and bounding boxes from the input models directly without an independent feature segmentation process. Therefore, the SsdNet can achieve a better runtime performance than the others. This is supported by the results illustrated in Table V. It is observed that the MsvNet takes 519.16 (=373.32 + 145.84) ms for feature segmentation and recognition, the FeatureNet takes 362.72 (=314.53 + 48.20) ms, and the SsdNet only takes 9.49 ms. It is also visible that the SsdNet and MsvNet take similar amount of times for postprecessing the outputs and selecting features.

As discussed previously, the MsvNet and FeatureNet are two-stage methods where the segmented features need to be passed through the networks separately. Therefore, the average number of segmented features and the average number of forward passes are identical in these approaches (see Table V). The SsdNet is an one-stage method where the average number of forward passes is a constant value (six forward passes per model). Therefore, the time complexity of the SsdNet is $\mathcal{O}(1)$.

F. Benefits Assessment of the SsdNet

As described in Section III, the SsdNet employs several training strategies, which could affect the final prediction performances. Therefore, the experiments under the following settings are conducted to examine the effects of these strategies: (1) The SsdNet with the default configuration suggested in previous sections is employed. In this experiment, TL and DA are enabled at the training stage. The output of the network is a set of 3-D bounding boxes. The learning rate is initially set as 10^{-4} and then changed to 10^{-5} at the third epoch. At the training phase, 512 models per feature type are utilized to train the networks. (2) In this setting, the output of neural network is a set of 2-D bounding boxes rather than 3-D boxes, which is identical to the original SSD algorithm [14]. The depth information of a potential feature is calculated based on a heuristic estimation method suggested in [13]. In this estimation, the depth of a bounding box is set to the maximal depth of all features appeared in this 2-D box. The rest configurations are identical to those in the previous setting. (3) The TL and DA in this setting are disabled during the training. (4) In this setting, the SsdNet with 2-D outputs is employed. The TL and DA are also disabled during training. (5) and (6) To examine the benefits of the learning rate decay strategy, the learning rates are set as fixed values in these two settings (10^{-4} and 10^{-5}, respectively). (7)–(10) To examine whether the proposed method is capable of producing satisfactory results when there are no sufficient 3-D models for training, the number of models per feature type utilized for training is set as 256, 128, 64, and 32, respectively in these four settings. (11) and (12) For the comparison purpose, the MsvNet and FeatureNet with the default settings are adopted as baselines. 512 models per feature type are employed for training.

TABLE VI
EXPERIMENTAL RESULTS (%) BASED ON DIFFERENT CONFIGURATIONS

	Method	TL & DA	Output	Learning Rate	#Models	$\mathcal{F}_r$	$\mathcal{F}_l$
(1)	SsdNet	✓	3D	$10^{-4}, 10^{-5}$	512	95.20	90.62
(2)	SsdNet	✓	2D	$10^{-4}, 10^{-5}$	512	89.64	78.32
(3)	SsdNet		3D	$10^{-4}, 10^{-5}$	512	91.30	86.45
(4)	SsdNet		2D	$10^{-4}, 10^{-5}$	512	86.01	74.38
(5)	SsdNet	✓	3D	10^{-4}	512	94.36	89.70
(6)	SsdNet	✓	3D	10^{-5}	512	93.34	88.93
(7)	SsdNet	✓	3D	$10^{-4}, 10^{-5}$	256	93.51	88.89
(8)	SsdNet	✓	3D	$10^{-4}, 10^{-5}$	128	90.44	85.54
(9)	SsdNet	✓	3D	$10^{-4}, 10^{-5}$	64	85.99	79.46
(10)	SsdNet	✓	3D	$10^{-4}, 10^{-5}$	32	84.29	77.76
(11)	MsvNet	-	-	-	512	76.24	58.26
(12)	FeatureNet	-	-	-	512	57.45	38.37

Table VI illustrates the recognition and localization F-score (denoted as $\mathcal{F}_r$ and $\mathcal{F}_l$, respectively) under the 12 experimental configurations. It is evident from the setting (1)–(10) that the SsdNet with the default configuration produces the best results in terms of feature localization and recognition. From the setting (1) and (2), it is observed that the network with 3-D outputs is better than that of 2-D outputs. This result indicates the deep learning algorithm outperforms the heuristic estimation method in calculating the depths of features. It is evident from the setting (1) and (3) that TL and DA could enhance the localization and recognition performances. This phenomenon can also be observed from the results captured from the setting (2) and (4). It is evident from the setting (1), (5), and (6) that the SsdNet with a simple learning rate decay strategy works better than the SsdNet with a fixed learning rate. Such an evidence is also supported by [33]. In the setting (7)–(10), it is observed that the number of 3-D models utilized for training could largely affect the final recognition and localization performances. The proposed method could achieve near-optimal results when 128–256 models per feature type are employed for training. In addition, the SsdNet with limited number of training samples (e.g., 32 models per feature type) could still produce better results than the MsvNet and FeatureNet, as evident in the setting (10), (11), and (12). From the setting (1), (11), and (12), it is visible that SsdNet with supervised feature segmentation method outperforms other approaches with unsupervised segmentation methods.

V. CONCLUSION

In conclusion, this article proposed a novel method for intersecting feature localization and recognition via SSD. A thorough evaluation was carried out to compare the proposed method to the others. Experimental results demonstrated that the SsdNet achieved the state-of-art localization and recognition performances on the benchmark test set due to the nature of supervised learning algorithm employed for feature segmentation. In addition, the training strategies adopted in this article considerably enhanced the recognition performances. Furthermore, the SsdNet was more efficient than others as it was a one-stage method. The proposed method could be utilized in a computer-aided process planning system, which produced a set of manufacturing operations and machine tools based on the feature localization and recognition results. With the insight obtained from this work, the feasibility in minimizing the required training samples and extending this approach to free-form features would be explored in the ongoing work.

REFERENCES

[1] J. Han, M. Pratt, and W. C. Regli, "Manufacturing feature recognition from solid models: A status report," *IEEE Trans. Robot. Autom.*, vol. 16, no. 6, pp. 782–796, Dec. 2000.

[2] B. Babic, N. Nesic, and Z. Miljkovic, "A review of automated feature recognition with rule-based pattern recognition," *Comput. Ind.*, vol. 59, no. 4, pp. 321–337, 2008.

[3] A. K. Verma and S. Rajotia, "A review of machining feature recognition methodologies," *Int. J. Comput. Integr. Manuf.*, vol. 23, no. 4, pp. 353–368, 2010.

[4] X. Xu, *Integrating Advanced Computer-Aided Design, Manufacturing, and Numerical Control*. Hershey, PA, USA: IGI Global, 2009.

[5] S. Shao, S. McAleer, R. Yan, and P. Baldi, "Highly accurate machine fault diagnosis using deep transfer learning," *IEEE Trans. Ind. Informat.*, vol. 15, no. 4, pp. 2446–2455, Apr. 2018.

[6] G. A. Susto, A. Schirru, S. Pampuri, S. McLoone, and A. Beghi, "Machine learning for predictive maintenance: A multiple classifier approach," *IEEE Trans. Ind. Informat.*, vol. 11, no. 3, pp. 812–820, Jun. 2015.

[7] R. C. Luo and C. W. Kuo, "A scalable modular architecture of 3-D object acquisition for manufacturing automation," in *Proc. IEEE 13th Int. Conf. Ind. Informat.*, 2015, pp. 269–274.

[8] C. Zhang, G. Zhou, H. Yang, Z. Xiao, and X. Yang, "View-based 3D CAD model retrieval with deep residual networks," *IEEE Trans. Ind. Informat.*, vol. 16, no. 4, pp. 2335–2345, Apr. 2020.

[9] R. C. Luo and C. W. Kuo, "Intelligent seven-dof robot with dynamic obstacle avoidance and 3-D object recognition for industrial cyber–physical systems in manufacturing automation," *Proc. IEEE*, vol. 104, no. 5, pp. 1102–1113, May 2016.

[10] H. Wang, Z. Li, Y. Li, B. B. Gupta, and C. Choi, "Visual saliency guided complex image retrieval," *Pattern Recognit. Lett.*, vol. 130, pp. 64–72, 2020.

[11] M. Al-Ayyoub, S. AlZu'bi, Y. Jararweh, M. A. Shehab, and B. B. Gupta, "Accelerating 3D medical volume segmentation using GPUs," *Multimedia Tools Appl.*, vol. 77, no. 4, pp. 4939–4958, 2018.

[12] Z. Zhang, P. Jaiswal, and R. Rai, "Featurenet: Machining feature recognition based on 3D convolution neural network," *Comput.-Aided Des.*, vol. 101, pp. 12–22, 2018.

[13] P. Shi, Q. Qi, Y. Qin, P. J. Scott, and X. Jiang, "A novel learning-based feature recognition method using multiple sectional view representation," *J. Intell. Manuf.*, vol. 31, no. 5, pp. 1291–1309, 2020.

[14] W. Liu *et al.*, "SSD: Single shot multibox detector," in *Proc. Eur. Conf. Comput. Vis.*, 2016, pp. 21–37.

[15] M. A. Alsmirat, F. Al-Alem, M. Al-Ayyoub, Y. Jararweh, and B. B. Gupta, "Impact of digital fingerprint image quality on the fingerprint recognition accuracy," *Multimedia Tools Appl.*, vol. 78, no. 3, pp. 3649–3688, 2019.

[16] Y. Liu and M. Zhu, "Processed RGB-D slam based on hog-man algorithm," *Int. J. High Perform. Comput. Netw.*, vol. 14, no. 3, pp. 376–384, 2019.

[17] B. R. Babic, N. Nesic, and Z. Miljkovic, "Automatic feature recognition using artificial neural networks to integrate design and manufacturing: Review of automatic feature recognition systems," *Artif. Intell. Eng. Des., Anal. Manuf.*, vol. 25, no. 3, 2011, Art. no. 289.

[18] J. H. Vandenbrande and A. A. Requicha, "Spatial reasoning for the automatic recognition of machinable features in solid models," *IEEE Trans. Pattern Anal. Mach. Intell.*, vol. 15, no. 12, pp. 1269–1285, Dec. 1993.

[19] T. Dipper, X. Xu, and P. Klemm, "Defining, recognizing and representing feature interactions in a feature-based data model," *Robot. Comput.-Integr. Manuf.*, vol. 27, no. 1, pp. 101–114, 2011.

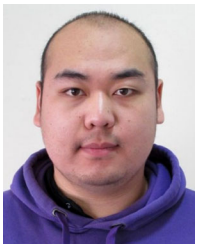
[20] A. Mokhtar and X. Xu, "Machining precedence of 21/2D interacting features in a feature-based data model," *J. Intell. Manuf.*, vol. 22, no. 2, pp. 145–161, 2011.

[21] Y. Woo, "Fast cell-based decomposition and applications to solid modeling," *Comput.-Aided Des.*, vol. 35, no. 11, pp. 969–977, 2003.

[22] Y. Li, Y. Ding, W. Mou, and H. Guo, "Feature recognition technology for aircraft structural parts based on a holistic attribute adjacency graph," *Proc. Inst. Mech. Eng., Part B, J. Eng. Manuf.*, vol. 224, no. 2, pp. 271–278, 2010.

[23] S. Xu, N. Anwer, and C. Mehdi-Souzani, "Machining feature recognition from in-process model of NC simulation," *Comput.-Aided Des. Appl.*, vol. 12, no. 4, pp. 383–392, 2015.

[24] G. Campana and M. Mele, "An application to stereolithography of a feature recognition algorithm for manufacturability evaluation," *J. Intell. Manuf.*, vol. 31, no. 1, pp. 199–214, 2020.
[25] L. Ding and Y. Yue, "Novel ANN-based feature recognition incorporating design by features," *Comput. Ind.*, vol. 55, no. 2, pp. 197–222, 2004.
[26] N. Öztürk and F. Öztürk, "Hybrid neural network and genetic algorithm based machining feature recognition," *J. Intell. Manuf.*, vol. 15, no. 3, pp. 287–298, 2004.
[27] E. Brousseau, S. Dimov, and R. Setchi, "Knowledge acquisition techniques for feature recognition in CAD models," *J. Intell. Manuf.*, vol. 19, no. 1, pp. 21–32, 2008.
[28] W. Liu, A. Rabinovich, and A. C. Berg, "Parsenet: Looking wider to see better," 2015, *arXiv:1506.04579*.
[29] R. Girshick, "Fast R-CNN," in *Proc. IEEE Int. Conf. Comput. Vis.*, 2015, pp. 1440–1448.
[30] M. Everingham, L. Van Gool, C. K. Williams, J. Winn, and A. Zisserman, "The pascal visual object classes (VOC) challenge," *Int. J. Delof Comput. Vis.*, vol. 88, no. 2, pp. 303–338, 2010.
[31] N. Bodla, B. Singh, R. Chellappa, and L. S. Davis, "Soft-NMS–improving object detection with one line of code," in *Proc. IEEE Int. Conf. Comput. Vis.*, 2017, pp. 5561–5569.
[32] C. Largeron, C. Moulin, and M. Géry, "MCut: A thresholding strategy for multi-label classification," in *Proc. Int. Symp. Intell. Data Anal.*, 2012, pp. 172–183.
[33] A. C. Wilson, R. Roelofs, M. Stern, N. Srebro, and B. Recht, "The marginal value of adaptive gradient methods in machine learning," in *Proc. Adv. Neural Inf. Process. Syst.*, 2017, pp. 4148–4158.
[34] L. Liu *et al.*, "Deep learning for generic object detection: A survey," *Int. J. Comput. Vis.*, vol. 128, no. 2, pp. 261–318, 2020.
[35] M. Sokolova and G. Lapalme, "A systematic analysis of performance measures for classification tasks," *Inf. Process. Manag.*, vol. 45, no. 4, pp. 427–437, 2009.



Peizhi Shi received the bachelor's degree in computer science from the Guilin University of Electronic Technology, Guilin, China, in 2010, the master's degree in software engineering from the University of Science and Technology of China, Hefei, in 2013, and the Ph.D. degree in computer science from the University of Manchester, Manchester, U.K., in 2019.

He is currently a Research Fellow with the EPSRC Future Advanced Metrology Hub, University of Huddersfield, Huddersfield, U.K. His current research interests include machine learning, 3-D object localization and recognition, machine perception, and their applications in intelligent system development.

Qunfen Qi received the Ph.D. degree in precision engineering from the University of Huddersfield, Huddersfield, U.K., in 2013.

She is currently a Senior Research Fellow with the EPSRC Future Advanced Metrology Hub, University of Huddersfield. She has worked for 15 years in developing decision-making tools for smart product design and inspection, using category theory as its foundation. Her research interests include knowledge modeling for manufacturing covering smart information systems, abstract mathematical theory (category theory), geometrical product specifications, additive manufacturing, and surface metrology.

Dr. Qi is an EPSRC UKRI Innovation Fellow, an EPSRC Peer Review Full College Member, an EPSRC Women in Engineering Society member, and a fellow of the Higher Education Academy.

Yuchu Qin received the bachelor's degree in computer science and technology and the master's degree in computer application technology from the School of Computer Science and Engineering, Guilin University of Electronic Technology, Guilin, China, in 2010 and 2013, respectively, and the first Ph.D. degree in measurement technology and instrument from the School of Mechanical Science and Engineering, Huazhong University of Science and Technology, Wuhan, China, in 2017. He is currently working toward the second Ph.D. degree in advanced manufacturing and precision engineering with the EPSRC Future Advanced Metrology Hub, University of Huddersfield, Huddersfield, U.K.

He has authored more than 30 papers in international journals such as *Virtual and Physical Prototyping, Knowledge-Based Systems, Robotics and Computer-Integrated Manufacturing, Journal of Intelligent Manufacturing, Computers and Industrial Engineering, Advanced Engineering Informatics, Computer-Aided Design*, and *Journal of Computing and Information Science in Engineering*. He has also coauthored two monographs about knowledge engineering. His research interests include intelligent manufacturing, computational intelligence, and knowledge engineering.

Paul J. Scott received the honours degree in mathematics, the M.Sc. degree in statistics, and the Ph.D. degree in statistics from the Imperial College London, London, U.K., in 1979, 1980, and 1983, respectively.

He is currently a Professor with the EPSRC Future Advanced Metrology Hub, University of Huddersfield, Huddersfield. He was the Project Leader for 20 published ISO standards and is currently working on four new ISO documents. His research interests include manufacturing informatics, geometrical product specifications and verification, philosophy of the measurement of product geometry, and foundations of specifying and characterising solutions for real world industrial problems.

Prof. Scott is a fellow of Royal Statistical Society, an EPSRC Fellow of Manufacturing, a leading member of ISO TC 213, a founder member of the strategic group AG1 and the technical review group AG2 of ISO TC 213, a convenor of the working group WG15 (Filtration and Extraction) and the advisory group AG12 (Mathematics for Geometrical Product Specifications) of ISO TC 213, a core member of BSI TDW4, a convenor of BSI TDW4/-/9, a Visiting Industrial Professor of Taylor Hobson Ltd, and the Taylor Hobson Chair for Computational Geometry.

Xiangqian Jiang received the D.Sc. degree in precision engineering from the University of Huddersfield, Huddersfield, U.K., in 2007, and the Ph.D. degree in surface metrology from the Huazhong University of Science and Technology, Wuhan, China, in 1995.

She is currently the Chair Professor and the Director of the EPSRC Future Advanced Metrology Hub, University of Huddersfield, Huddersfield, U.K. and the Royal Academy of Engineering and Renishaw Chair in Precision Metrology. She was made a Dame Commander (DBE) of the Order of the British Empire for services to Engineering and Manufacturing in 2017. Her research interests include surface measurement, precision engineering, and advanced manufacturing technologies.

Prof. Jiang is a fellow of the Royal Academy of Engineering, a fellow of the Royal Society of Arts, a fellow of the Institute of Engineering Technologies, a fellow of the International Academy of Production Research, a fellow of the International Society for Nanomanufacturing, a principle member of ISO TC 213 and BSI TW/4, an advisory member for UK national measurement system, and the UK Chairman of the International Academy of Production Research.