IEEE Access
Multidisciplinary ⋮ Rapid Review ⋮ Open Access Journal

Received February 18, 2020, accepted April 2, 2020, date of publication April 21, 2020, date of current version May 4, 2020.

Digital Object Identifier 10.1109/ACCESS.2020.2989300

End-Face Localization and Segmentation of Steel Bar Based on Convolution Neural Network

YONGJIAN ZHU, CHULIU TANG, HAO LIU, AND PENGCHI HUANG
College of Electronic Engineering, Guangxi Normal University, Guangxi 541004, China

Corresponding author: Chuliu Tang (iloveitre@gmail.com)

This work was supported in part by the National Natural Science Foundation of China under Grant 51775230, in part by the Natural Science Foundation of Guangxi Zhuang Autonomous Region under Grant 2017GXNSFAA198313 and Grant 2018GXNSFAA294003, and in part by the Cultivation Plan for 1000 Young and Middle-Aged Key Teachers from Universities of Guangxi Zhuang Autonomous Region.

ABSTRACT Both number manually-counting method and traditional Machine-Vision (MV) number counting strategy are laborious and very time-consuming (sometimes several hours). Thus a new deep learning (DL) fusion model is proposed, which includes object detection and semantic segmentation. It can solve the problems of end-face localization and segmentation of steel bars at the same time. In this fusion model, firstly, an improved data augmentation method namely, Sliding Window Data Augmentation (SWDA) is adopted to compensate less training data concerning object detection, based on which a new object-detection architecture, Inception-RFB-FPN is presented to improve the accuracy and inference time. Secondly, a novel AI labeling method, Fibonacci-incremental mask labeling method (FIMLM) is introduced to accelerate the generation of annotation mask. Furthermore, by contrast, three FCN (Fully Convolutional Networks) architectures of data segmentation, namely, VGG16-FCN, ResNet18-FCN, and ResNet34-FCN are used to conduct the end-face segmentations of steel bars separately. Finally, a series of experiments show that the proposed Inception-RFB-FPN model can reach 98.17% in F1 score (harmonic mean value of precision and recall) with respect to object detection, and its inference time only needs 0.0306 seconds, much faster than some related reports. In addition, the FIMLM-based ResNet34-FCN model can reach 97.47% in mean Intersection-Over-Union (mIOU) with respect to semantic segmentation, higher than both VGG16-FCN and ResNet18-FCN.

INDEX TERMS Steel bar, data augmentation, object detection, semantic segmentation.

I. INTRODUCTION

With the development of city, the required number of steel bars is becoming larger and larger at the building site, which is indispensable to support the constructure. But it's difficult and even annoying to count the number of steel bars manually. Workers usually need to spend several hours to count the steel bar's number. With the coming Industrial 4.0, AI is widely applied to industry, even in the counting process of steel bars at the building site. In AI application, the automatic localization and segmentation of steel bars are needed at first after achieving the images of a bundle of steel bars. On the other hand, a big problem needs to be tackled, which lies in the trade-off between accuracy and consuming time. Therefore, the deep learning (DL) method is used to realize the localization and segmentation of the steel bar's end face.In this paper, we want to study and rethink the augmentation method of small object localization, taking the end-face segmentation of steel bars into consideration. In particular, we find a data augmentation solution for small-object localization in case of inadequate data, which can provide more training data for object detection to reach better accuracy and less inference time. In the early study, a small object is hard to be detected in the image because of too large downsample ratio. Although it can be solved by reducing the downsampling ratio, there's a big problem that there exists only a small object number in dataset. An easy way to tackle this problem is to fill k times of small object around target localization in an image. However, it is not suitable for the augmentation dataset. The diversity of small objects can improve the small object detection accuracy. Based on this observation we propose a small object augmentation method,

The associate editor coordinating the review of this manuscript and approving it for publication was Xi Peng.

namely, Sliding Window Data Augmentation (SWDA). Our experiment shows that the proposed Inception-RFB-FPN can achieve a better accuracy and less inference time.

The main contributions to the paper can be summarized as follows:

(1) We improve a small data augmentation method which can be used as a data-processing strategy to achieve better performance in image localization.

(2) We propose a Fibonacci-incremental mask labeling method, which works well in Segmentation dataset.

(3)By use of RFB block of FPN, we design a steel bar localization architecture Inception-RFB-FPN to improve the localization accuracy, and to save the inference time.

(4) SWDA-CNN is composed of Inception-RFB-FPN and modified ResNet34-FCN, which can be used for steel bar localization and segmentation.

Then this paper is organized as follows: Section II, the challenges and related works are briefly discussed; Section III, the proposed method is described; Section IV, a series of experiments are conducted to prove our methods. Section V, the summaries are given.

II. RELATED WORKS

Nowadays, the present DL method faces many difficulties and challenges in processing the steel bar's image about number counting. Because the images captured from the building site are different depending on the on-site conditions, there exist some problems such as irregular end shapes, uneven illuminance, non-uniform colors, and overlapped end faces, etc. All of these factors lead to an unstable recognizing result when using the present DL image-processing algorithms, which often requires lots of data to be trained. In fact, there are usually no enough data to join the training group. Therefore, a special data augmentation method is needed to solve this problem. After the data augmentation is finished, two kinds of operations need to be introduced to realize the number counting of steel bars, namely end-face localization and segmentation.

In end-face localization, besides the manual counting, some traditional Machine-Vision (MV) methods have been adopted, Luo and Li [1] have proposed a K-level fault tolerance method and used the bidirectional linked lists to achieve the steel bar location and its offset; Similarly, in [2], Wu *et al*. have improved this method by use of concave point segment. In [3], Zhang *et al*. have proposed a template matching and mutative threshold method to implement the on-line steel bar counting and automatic separating system. In [4], Ying *et al*. have combined Sobel operator and Otsu to get the foreground and used Hough transform to enhance gray values in order to localize the steel bars. In [5], Su *et al*. have adopted the modified gradient Hough circle transform to localize the steel bars. In [6], Wang *et al*. have proposed a new segmentation method based on a quasi-circular assumption to count the Bounded steel bars. In [7], Nie *et al*. have used the matching algorithm to identify the adjacent frames and record the moving steel bars. In [8], Ghazali *et al*. have applied the Hough transform and a series of morphological operations to get the circle and rectangle shape of steel bar. In [9],Yan *et al*. have proposed a single-multi-classification of connected regions based on feature matching. However, all these MV-based methods are easily affected by light illumination and machine jitters, so some algorithm parameters need to be adjusted in real-time. In addition, Liu *et al*. [10]. adopt HOG features to locate steel bar center based on a machine learning SVM (support vector machine) classifier; Fan *et al*. [11] propose a CNN model named CNN-DC to achieve a high-accuracy counting rate (99.26 %) and localization simultaneously, but they need a long inference time of about 3.5 seconds.

In end-face segmentation, Arbeláez *et al*. [12] have proposed a region-based semantic model to implement pixel segmentation. In [13], Kampffmeyer *et al*. have modified the FCN by median frequency balancing method and achieved the high accuracy and F1 score. In [14], Noh *et al*. have proposed a convolution and deconvolution network to generate dense semantic masks. Lin *et al*. [15] have also proposed a patch-patch combined CRF (condition random fields) to avoid overlong CRF inference time due to iterative optimization. Reference [16] has proposed a generic multi-path refinement network to execute the high-resolution prediction. In [17], Zhao *et al*. have mentioned a feature pyramid pooling framework and achieved 85.4% of Intersection-Over-Union(mIOU). These DL methods above are mainly based on the deeper neural network and high-resolution image. In [18], it has proposed a method based on a geodesic distance-based technique for video segmentation, but the optical flow computation is very time-consuming. In [19],video object detection has been implemented via FCN, they adopted double FCN models, which can only reach 2 FPS for a single image(224×224).In [20],they used the pyramid attention and salient edge for object detection with 25 FPS/image. In [21],an iterative and cooperative FPN is used in object detection, but it's not suitable for the on-site complex situation. Likewise, Protonet [22] also faced a mask Leakage problem after being cropped in overlapped object detection, it can't even separate the very close objects. So inspired by Protonet work, in this paper, we build a new tiny FCN to reduce the large bounding-box crops for very-close objects and thus to save the inference time.

We propose a data augmentation method, namely, Sliding Window Data Augmentation (SWDA) to train an advanced deep-learning detector for counting the number of steel bars. After the location of each steel bar is confirmed in the DL model, small patches (128×128 pixels) are cropped by Numpy slice operation, which are then fed into FCN (Fully Convolutional Networks) model. Thus semantic masks are acquired. In this case, a data augmentation method is used to generate more data in order to improve the recall and precision rates. Then, the bounding box of each steel bar area is cropped and resized to be 128×128 pixels. Afterward, three FCN models are used to conduct the high-quality semantic

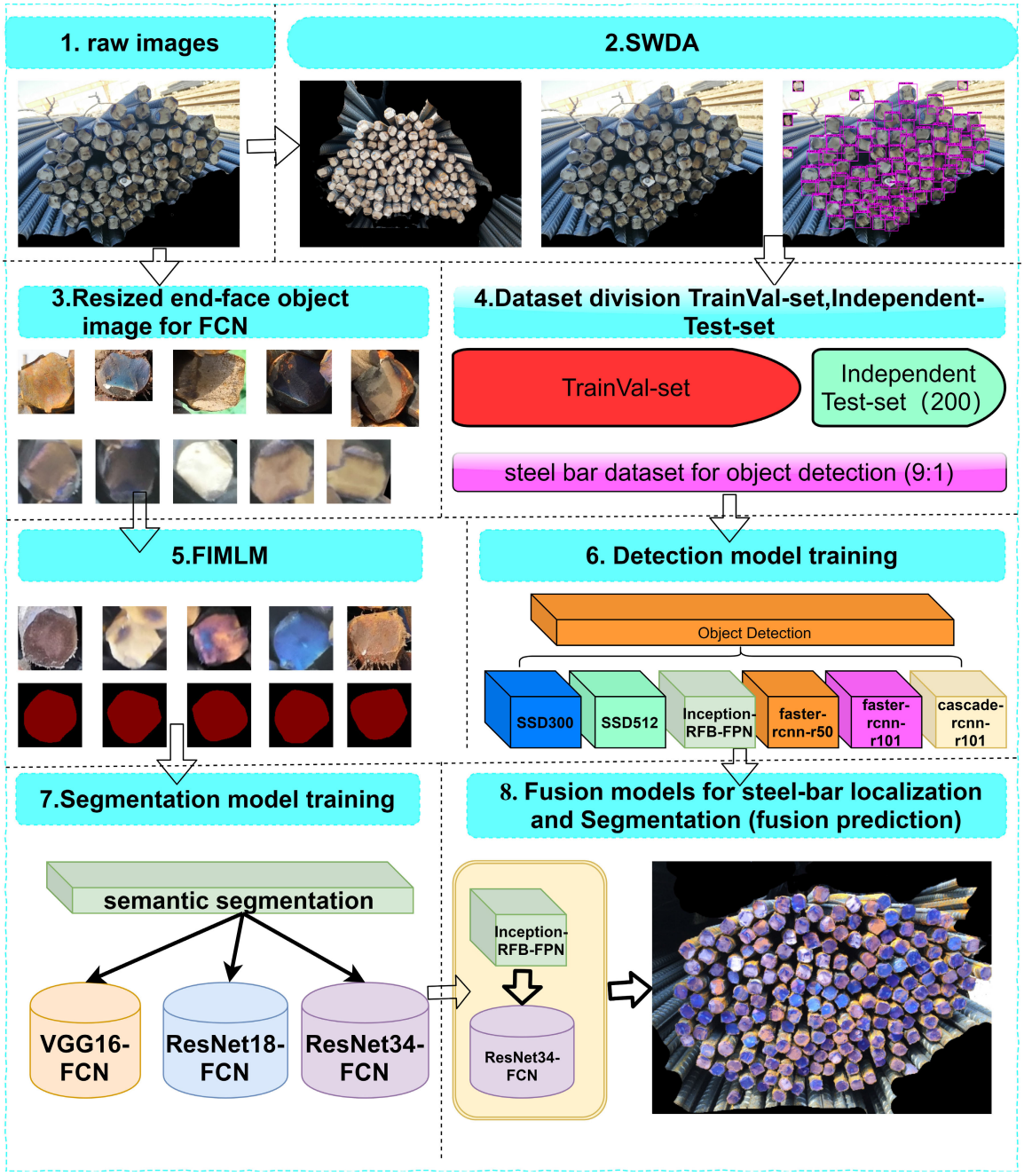


FIGURE 1. The Framework of proposed SWDA-CNN localization and segmentation of steel bar's end face.

segmentation for the end face of steel bar, at last, we combine DL detector and FCN models by Numpy and OpenCV to acquire the final results.

III. PRINCIPLE OF METHOD

A. SEGMENTATION FRAMEWORK OF STEEL BAR'S END FACE

As shown in Fig.1, we propose a framework based on the SWDA-CNN localization and segmentation of the steel bar's end face, which includes SWDA, Object Detection, FIMLM (Fibonacci-incremental mask labeling method), Semantic Segmentation and result mapping.

In Fig.1, there are two sections. Section one(object detection) includes (1),(3),(5),(7).Section two(semantic segmentation) includes (1),(2),(4),(6).Final results can be predicted by Inception-RFB-FPN and ResNet34-FCN.

(1) In all, there are 250 steel bar's raw images from [23], with 2666×2000 pixels in resolution. These images captured from the building site are different depending on the on-site

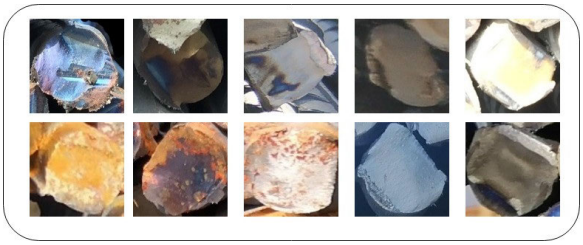


FIGURE 2. Cropped images of End face of steel bar (128 × 128 pixels).

conditions, there exist some problems such as irregular end shapes, uneven illuminance, non-uniform colors and overlapped end faces, etc.

(2) A new data augmentation method SWDA is proposed to solve the problem of less data, which can't drive the DL model.

(3) To meet the training requirement in a semantic segmentation model (FCN), each object is cropped from 250 raw images. Because the CNN model usually needs an input of square size, we resize the width and height of all objects to be 128 × 128 pixels.

(4) When the data augmentation is completed, the augmented dataset is randomly divided by ratio of 8:1:1 for training model of object detection.

(5) For the supervised DL, the FCN model needs lots of labeled images, but manually labeling work is laborious. So an improved labeling method is proposed based on FIMLM, which is to aim at getting a high mIOU.

(6) Comparison of various object detection models, namely SSD300, SSD512, Faster-rcnn-r50, Faster-rcnn-r101, Cascade-rcnn-r101 and the proposed Inception-RFB-FPN.

(7) Comparison of three semantic segmentation models, namely VGG16-FCN, ResNet18-FCN, and ResNet34-FCN.

(8) The object detection model produces bounding boxes, based on which raw images are cropped and resized to be 128 × 128 pixels in order to meet the requirement of FCN input. Finally, bounding boxes and masks are mapped back into the raw images.

B. SWDA AND FIBONACCI-INCREMENTAL MASK LABELLING METHOD (FIMLM)

1) SWDA

Fig.2 shows the appearance characteristics of the steel bar's end face. At the site, it's difficult to acquire high-quality raw images, which bring some troubles to model training. Inspired by the works of Kisantal *et al*. [24], we adopt an improved data augmentation method for the small object detection, namely SWDA. The procedure is described in algorithm1.

After SWDA, the object number in each image increases by K times compared to the original dataset of steel bar. Then the less-data problem can be solved by SWDA.

Algorithm 1 Algorithm of SWDA

Input: sample number K,image$A$,label$A$,image$B$,label$B$.
Output: augmentation image$A$,label$A$.

1: Read sample filenames recorded in a list.
2: Randomly shuffle the list.
3: From the list Pops two element sample image names,A and B.
4: Read image(A and B) and label(A and B).
5: label lists C and bounding boxes (BBox) D belong to label$A$.
6: **for** each i in BBox list D **do**
7: set all i region values to be 255
8: **end for**
9: **for** $step = 0$ to K **do**
10: get object and label from image$B$,label$B$.
11: use sliding window to generate valid area
12: add object and label to augmentation image$A$,label$A$.
13: **end for**
14: **return** augmentation image$A$,label$A$.

Main Steps of SWDA,

(1) SWDA reads an image and annotations(Algorithm line 1-5), shown in Fig3, SWDA transfers all the annotated files to the folder, stores them in a list, and then disorders the list. We will pop up two filenames from this list. According to these two filenames, we will read the image A and its corresponding annotation label A, image B and label B. label A includes label list C and bounding box list D, and label B is also the same.

(2) SWDA generates object masks(Algorithm line 6-8), and object region value is set to be 255, shown in Fig4. In order to avoid the occlusion the original image data, we generate a mask area to facilitate the subsequent generation of valid areas.

(3) According to object mask by using Sliding Windows method, SWDA generates valid area (namely green box area, including no value 255), and adds new object and label, then the positions are randomly selected from valid areas (for example K = 3).(Algorithm line 9-13),shown in Fig5. It is noted that the sliding window is generated by Python yield function.

(4) SWDA returns image and label (Algorithm line 9-13).

2) FIMLM

Labeling operation is very time-consuming for the traditional supervised learning method due to the high resolution (2666 × 2000 pixels) of raw images with many objects. At present, there are many labeling tools such as Label Me and Labelimg, but they are very laborious and not efficient. So Castrejón [25] has proposed a labeling tool PolyRNN, then Acuna *et al*. [26] has modified PolyRNN to be PolyRNN++, which becomes an automatic labeling tool for semantic segmentation. PolyRNN and PolyRNN++ aim to predicting the convex point of object, but generally, the steel bar has an

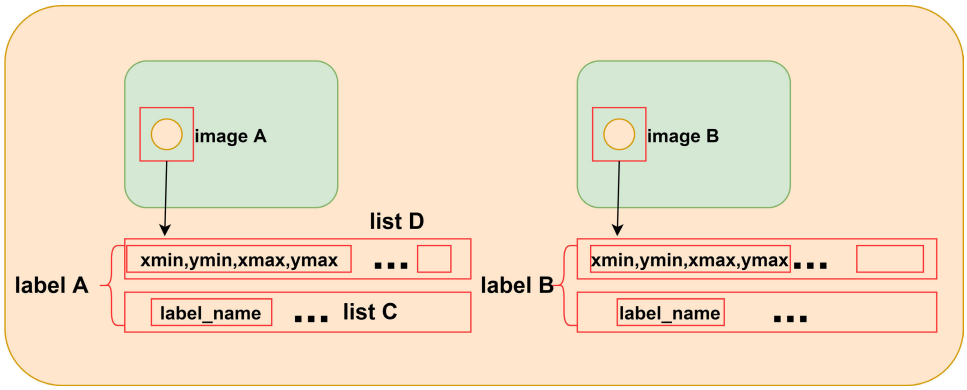


FIGURE 3. Read image A, B and label A,B.

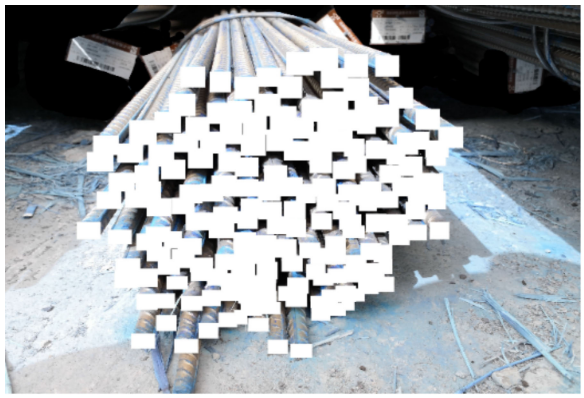


FIGURE 4. SWDA generates object mask.

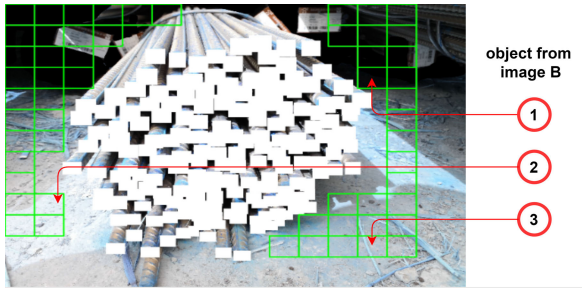


FIGURE 5. SWDA generates valid area and adds object, label.

irregular end face, it is very hard to find the convex point. Accordingly, we propose a new semi-automatic labeling method, namely FIMLM. In Photoshop software, the annotation labels adopt VOC data format and set the color panel background to be (0,0,0), steel bar to be (128,0,0). FIMLM includes the following steps: firstly, make manual annotations of 300 patches (cropped objects from raw image, all resized to be 128 × 128 pixels suitable for ResNet34-FCN model input); secondly, train a semantic segmentation model and evaluate the model by mIOU; thirdly, when the first model is trained, we choose a Fibonacci series as the incremental ratio. In Fibonacci series $1, 1, 2, 3, 5 \ldots$, the first number is abandoned, and the left numbers are multiplied by 300, so the annotation number series is $300, 600, 900, \ldots n, n+1$, finally, the FCN model is iterated repeatedly until mIOU reaches a good value.

Figure 6 shows the operation process of FIMLM. The first-round 300 images are made by manual annotation, and we use the first-round dataset to train Resnet34-FCN model. when the first-round training is completed, the trained Resnet34-FCN (semantic model) is used to predict 600 unlabeled images. At the beginning, the semantic model doesn't work very well because it predicts some wrong masks. Here we keep the qualified masks, and abandon the wrong masks. All the qualified masks in the first-round prediction will mix up with previously trained dataset, for example, the second-round dataset includes the first-round data and first-round qualified masks. In this process, man only picks up the wrong masks without any subjective annotation. we repeated the above process and iterate semantic model until the fifth round dataset is trained. After we have a very good test result of mIOU, we use a semantic model to predict all the rest unlabeled images.

C. INCEPTION-RFB-FPN ARCHITECTURE

A series of baseline models are trained for the localization of steel bars. Liu *et al* [27], have proposed Single Shot Multibox Detector(SSD), and Ren *et al*. proposed Faster-RCNN [28]. To acquire the results of baseline models quickly, Mmdetection toolbox [29] is used to train and test these models based on standard parameters, and then to analyze these results.

Inception-RFB-FPN model is used for localization. Inception-RFB-FPN model is implemented based on open-source framework PyTorch, fundamental code of RFBNet [30] and open-source repository [31]. Same to RFBNet, the training steps include online data augmentation, hard negative mining, and calculation of loss function (localization by Smooth L1 Loss and classification by Softmax Loss). Fig.7 shows the Inception-RFB-FPN architecture.

In Fig.7, the head layer is stacked by three convolution layers, and a series of Inception blocks with pooling layer (abstract layers) extract the foreground information.

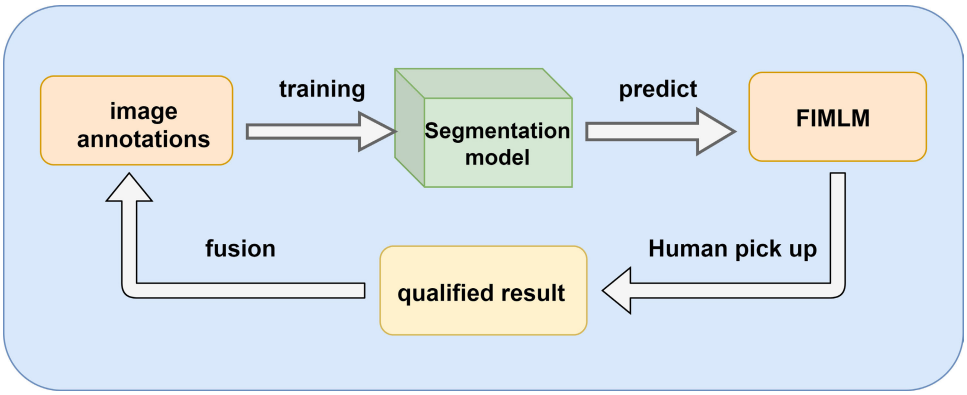


FIGURE 6. Operations of FIMLM.

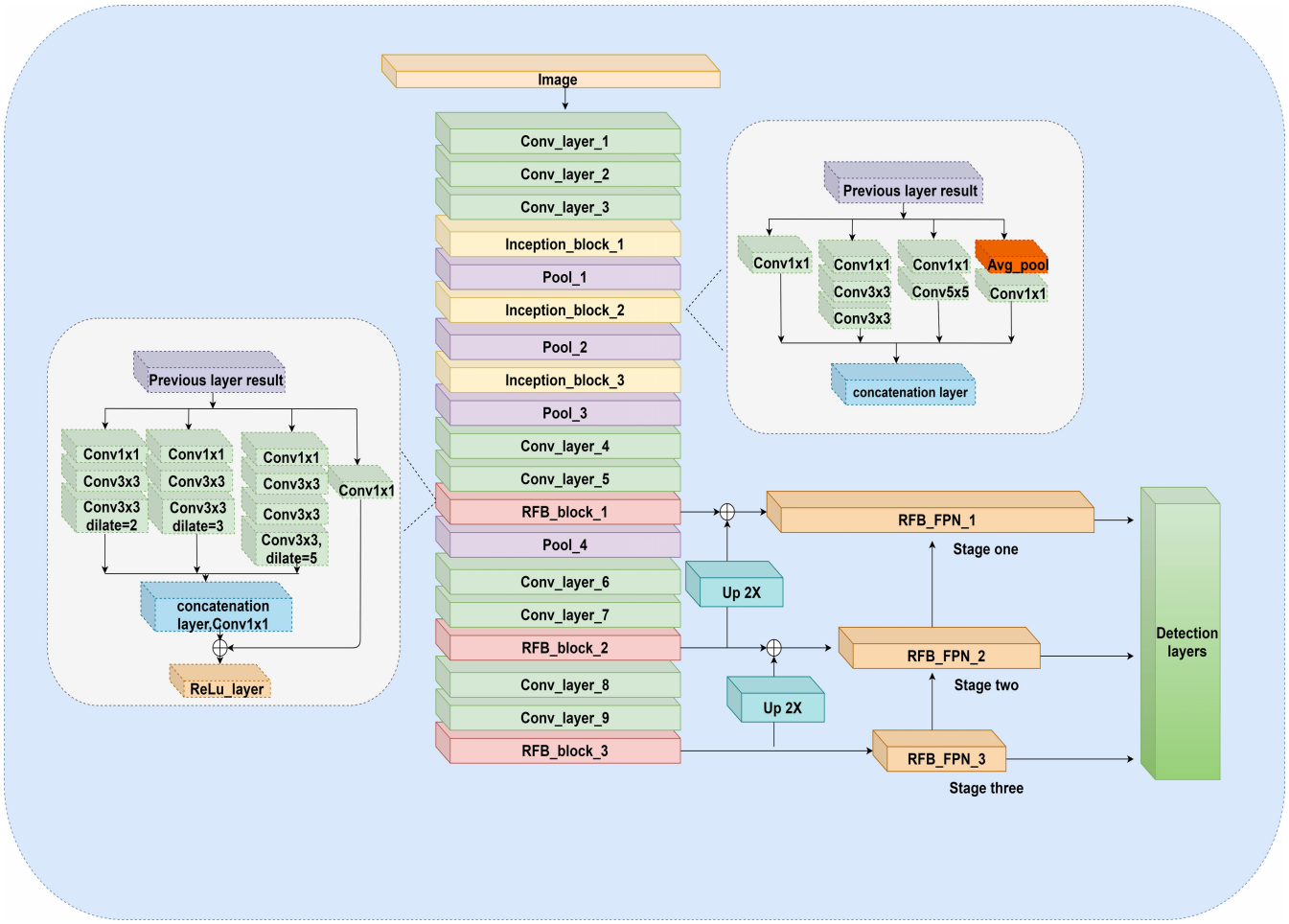


FIGURE 7. Inception-RFB-FPN architecture.

To improve the model's generalization ability, we stack three Inception blocks to generate the 20-layer Inception-RFB-FPN neural network, which is powerful and time-saving. However, we don't stack more layers because a deep neural network has an explosive parameter increasement. Then an RFB-FPN architecture is designed to recover the object feature from high-level feature maps. Detection layers are to acquire the BBox and object probability following the RFB-FPN stage. In Head layers, three Conv3 $\times$ 3 with 32 channels are used, then the Inception block has four branches, namely, branch one: Conv1 $\times$ 1 with 32 channels; branch two: Conv1 $\times$ 1 with 32 channels and double Conv3 $\times$ 3

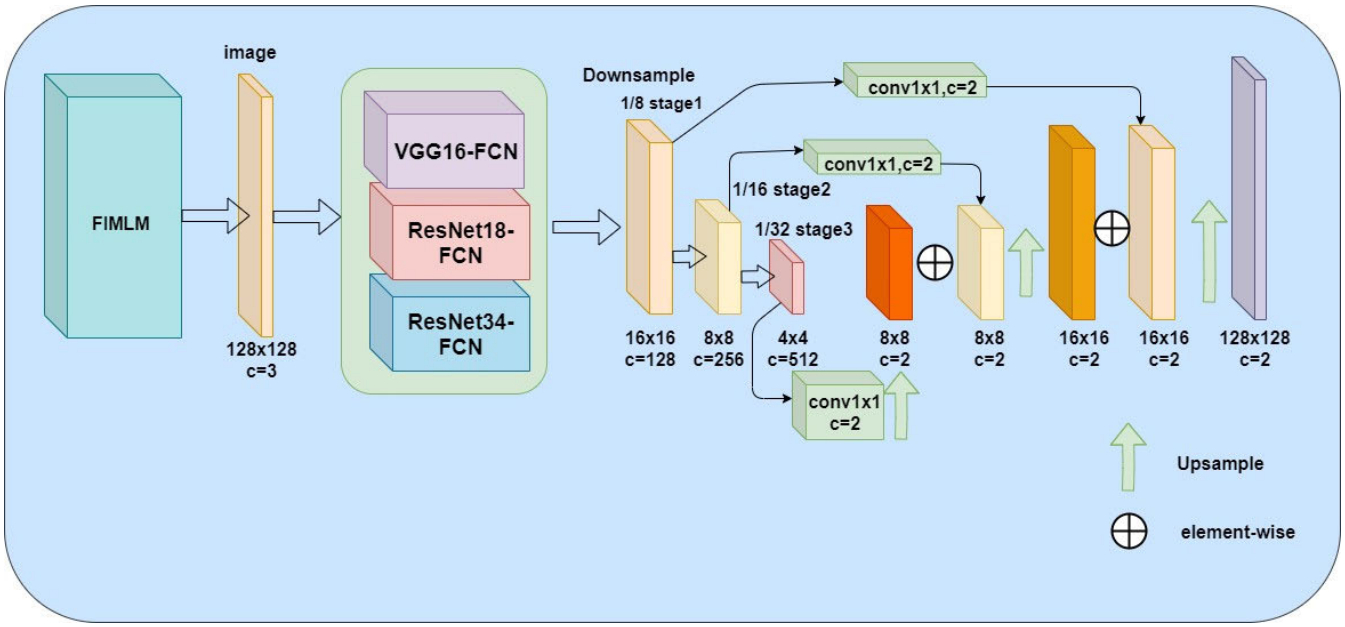


FIGURE 8. FCN model architecture with small input size.

with 64 channels; branch three: Conv1 × 1 with 32 channels and Conv5 × 5 with 64 channels; branch four: average pooling 3 × 3, stride 1 and Conv1 × 1 with 32 channels. After that, a Max-pooling layer 2 × 2 with stride 2 is used to reduce the feature map size. Afterward, the Inception block with the Max-pooling layer will be repeated three times. At last, RFB-FPN is constructed by two convolution layers and an RFB block. From Fig.5, stage one consists of Conv1 × 1 with 128 channels, Conv3 × 3 with 128 channels and RFB-one: stack Conv1 × 1, Conv3 × 3, Conv3 × 3 with dilation 2,3, and 5 in succession. Stage two and Stage three are similar to Stage one except for Conv1 × 1 with 64 channels. Both Stage two and Stage three have 2X up-sampling (binary interpolation) and element-wise addition to merge the previous feature maps. Scales and aspect ratios are set to be 20, 40 and 60 separately and to be 0.9. All convolution layers are initialized by KaiMing-norms.

D. FCN ARCHITECTURE

The FCN model is used for segmentation. Because FCN model [32] costs a long inference time due to the adoption of a large-size input image, a small input one is needed to improve the model performance. In Fig.8, the small input FCN model is shown, in which a small image patch is provided by 3X down-sampling of the feature map and by deconvolution layers (initialized by bilinear kernel method). In this case, the down-sampling feature maps are derived from one of these three backbones, namely, VGG16, ResNet18, and ResNet34. Then, the up-sampling feature maps are achieved by deconvolution layers. Consequently, the down- and up-sampling feature maps are superposed by element-wise addition to get the final feature maps. Finally, the loss value is calculated pixel by pixel by soft-max cross-entropy. In three FCN architectures, ResNet34-FCN performs best.

TABLE 1. Computer configuration.

Hardware platform:
CPU:4 core
Memory:30GB
GPU:NVIDIA T4 16G memory
Software platform:
System:Ubuntu16.04 LTS
Code Edit: Pycharm with Python3.6
Deep learning framework PyTorch1.2.

IV. EXPERIMENTAL AND RESULTS

A. EXPERIMENT CONFIGURATION

Computer configuration is shown in Table 1:

The training dataset contains object detection and semantic segmentation. As shown in Table 2, object detection dataset (raw dataset) comes from Data Fountain platform [23], which contains 250 train images and 200 test images. In Table 2, the traditional augmentation dataset has 1350 images, and the SWDA dataset contains 2000 images. We divide all datasets by ratio of 8:1:1 into train set, validate set and test set. These datasets comprise various images of irregular end-face shape, uneven illuminance, non-uniform color and overlapped end-face. All of these factors will lead to an unstable detection result. When training the models, we use the trainval (train and validate) set. The split test dataset is abandoned. We use an independent test dataset (200) to testify the

TABLE 2. Object detection data set.

Types	Raw dataset	Traditional raw augmentation	SWDA
Train set(image number)	200	1080	1600
Validation set(image number)	25	135	200
Individual test set(image number)	200	200	200
Total image number	425	1415	2000

TABLE 3. Semantic segmentation data set.

Types	Round1	Round2	Round3	Round4	Round5	Round6
Train set(image number)	240	648	1285	2420	4192	23166
Validation set(image number)	30	81	161	302	524	2896
Test set(image number)	30	80	160	302	523	2895
Total image number	300	809	1606	3024	5239	28957

model's performance. Generally, the traditional augmentation method adopts the Imageaug Python library including some operators such as Vertical Flip, Mirror, Brightness, Gaussian blur and Affine.

Table 3 shows the semantic segmentation dataset, which comes from the raw dataset of steel bar. The objects are cropped and resized to be 128 × 128 pixels. After FIMLM, man manually picks up the qualified sample images and labels, and merges them round by round (training stage), in all, there are six rounds from Round1 to Round6. All Rounds of datasets are divided by ratio of 8:1:1. In Round6, there are 23166 images treated as the training set, 2896 images as the validation set and 2895 images as the test set.

B. MODEL TRAINING AND EVALUATION

1) END FACE LOCALIZATION OF STEEL BAR

Open source framework Mmdetection is used to get the localization of end face, in which the object detection models adopt SSD300, SSD512, Faster-rcnn-r50, and Faster-rcnn-r101, respectively. To evaluate the object detection models, Recall, Precision, and F1 score are used. Recall, Precision, and F1 score are calculated by the following equations from (1) to (3).

$$Recall = \frac{TP}{TP + FN} \tag{1}$$

$$Precision = \frac{TP}{TP + FP} \tag{2}$$

$$F1 = 2 \times \frac{Recall \times Precision}{Recall + Precision} \tag{3}$$

where TP is true positive, FP false positive, FN false negative, and TN true negative.

As shown in Table 4, the traditional data augmentation has little lower scores in Recall, Precision, and F1 than the non-augmentation method, that is to say, data augmentation method does work.

To further improve the localization accuracy, a new SWDA is considered. By calculating, Recall, Precision, and F1 score are achieved as shown in Table 6.

At the same time, the inference time of each method is calculated and shown in Table 4,5 and 6. All these models perform the testing in Table2 by use of individual test dataset. Furthermore, the proposed method is compared with the results of other references, which are shown in Table 7. It denotes that our proposed Inception-RFB-FPN model reaches F1:98.20% and needs the least inference time.

TABLE 4. In SSD300, SSD512, Faster-rcnn-r50, Faster-rcnn-r101, and Cascade-rcnn-r101 models, non-augmentation method.

model	Recall	Precision	F1 Score	Times(s)
SSD300	0.828	0.785	0.806	0.0686
SSD512	0.944	0.936	0.940	0.0968
Faster-rcnn-r50	0.708	0.707	0.707	0.1356
Faster-rcnn-r101	0.708	0.706	0.707	0.1493
cascade-rcnn-r101	0.707	0.706	0.706	0.4461

TABLE 5. In SSD300, SSD512, Faster-rcnn-r50, Faster-rcnn-r101, and Cascade-rcnn-r101 models,traditional augmentation method.

model	Recall	Precision	F1 Score	Times(s)
SSD300	0.853	0.812	0.832	0.0686
SSD512	0.926	0.906	0.916	0.0968
Faster-rcnn-r50	0.701	0.696	0.698	0.1356
Faster-rcnn-r101	0.706	0.704	0.705	0.1493
cascade-rcnn-r101	0.707	0.705	0.706	0.4461

TABLE 6. Cascade-rcnn-r101,Inception-RFB-FPN SWDA dataset with different k number result.

models name	Recall	Precision	F1 Score	Times(s)
cascade-rcnn-r101(k=1)	0.707	0.705	0.706	0.4461
cascade-rcnn-r101(k=3)	0.708	0.706	0.707	0.4461
cascade-rcnn-r101(k=5)	0.707	0.705	0.706	0.4461
cascade-rcnn-r101(k=7)	0.708	0.705	0.706	0.4461
cascade-rcnn-r101(k=9)	0.707	0.705	0.706	0.4461
Inception-RFB-FPN(k=1)	0.9874	0.9716	0.9794	0.0306
Inception-RFB-FPN(k=3)	0.9869	0.9718	0.9793	0.0306
Inception-RFB-FPN(k=5)	0.9881	0.9747	0.9814	0.0306
Inception-RFB-FPN(k=7)	**0.9881**	**0.9753**	**0.9817**	**0.0306**
Inception-RFB-FPN(k=9)	0.9879	0.9744	0.9811	0.0306

TABLE 7. Compared with the results from other references.

Method	Recall	Precision	F1 Score	times(s)
Zhang *et al.* [3]	0.886	0.936	0.910	0.3023
Ying *et al.* [4]	0.962	0.842	0.898	0.2404
Ghazali *et al.* [8]	0.978	0.937	0.957	0.1346
Liu *et al.* [35]	0.812	0.683	0.742	0.0313
Fan *et al.* [11]	**0.995**	**0.998**	**0.992**	3.5862
Proposed(k=7)	0.988	0.975	**0.982**	**0.0306**

2) DETECTION RESULT DISCUSSION AND VISUALIZATION

The test dataset comprises 200 images with different scales, occlusion and illumination. From Fig 9,10,11,and12,we find that without augmentation dataset, the traditional augmentation method can't reach a good F1 score, but SWDA by use of augmentation dataset can improve Inception-RFB-FPN to reach a higher F1 score. In SWDA K number experiment, we search in the range [1,3,5,7,9]. Results show K number must be over the Nyquist Sampling Theorem rate. In our experiment, the model has the highest F1 score when K is 7. Some object detection results are on the test dataset by using

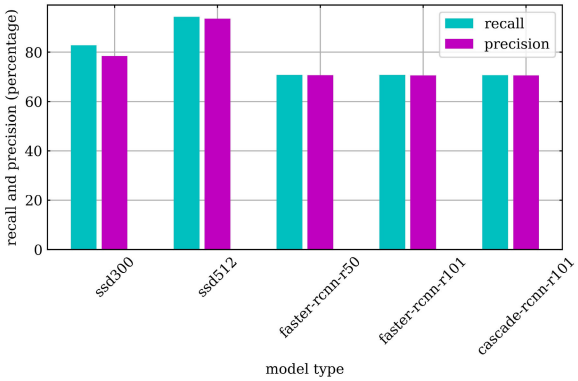


FIGURE 9. Non-augmentation method recall precision trade off.

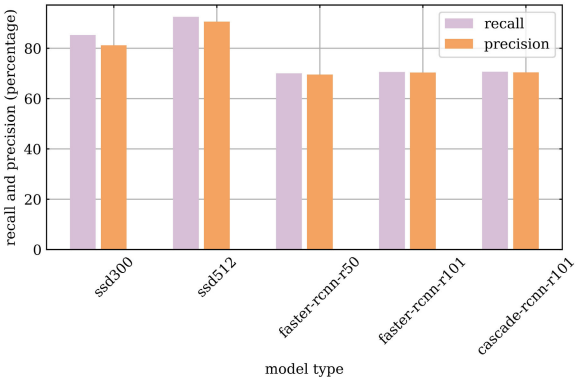


FIGURE 10. Traditional augmentation method recall precision trade off.

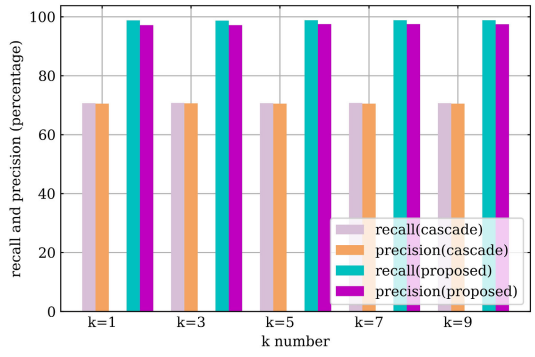


FIGURE 11. SWDA augmentation method recall precision trade off.

Inception-RFB-FPN (SWDA k = 7), on these images the yellow bounding box represents normal detection, and the red box is error detection or target loss. From Fig13, the proposed model leads to some confusions due to occlusion, strong illumination and truck wheel, but it has a good compatibility of scale, irregular end shape and non-uniform color (normal detection). Owing to the different view angles of hand-held mobile phone, there is a certain degree of physical occlusion that targets are partly missing. It is inevitable that

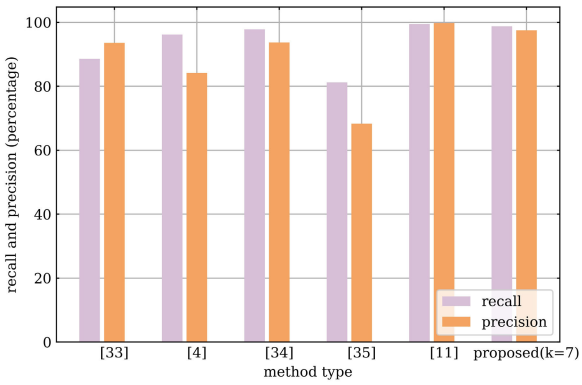


FIGURE 12. Compare with other reference method recall precision trade off.

TABLE 8. Properties of semantic segmentation model.

Model name	Training types	Number layers	FLOPS	Filter parameter number
VGG16-FCN	scratch	16	5590585344	14718406
VGG16-FCN	fine-tuned	16	5590585344	14718406
ResNet18-FCN	scratch	18	613158912	11179462
ResNet18-FCN	fine-tuned	18	613158912	11179462
ResNet34-FCN	scratch	34	1218981888	21287622
ResNet34-FCN	fine-tuned	34	1218981888	21287622

there will be some error detections, In the case of small occlusion, the proposed model performs well. From the detection results, the wheel is detected by the proposed model because the truck wheel has a similar circle shape. In the open environment, the background of image may be affected by the sky. In the case of strong illuminance condition, the steel bar's boundary and the object color will become blur, resulting in the loss of target. The steel slags scattered on the ground are also considered as the targets because their colors are very close to the target.

3) END FACE SEGMENTATION OF STEEL BAR

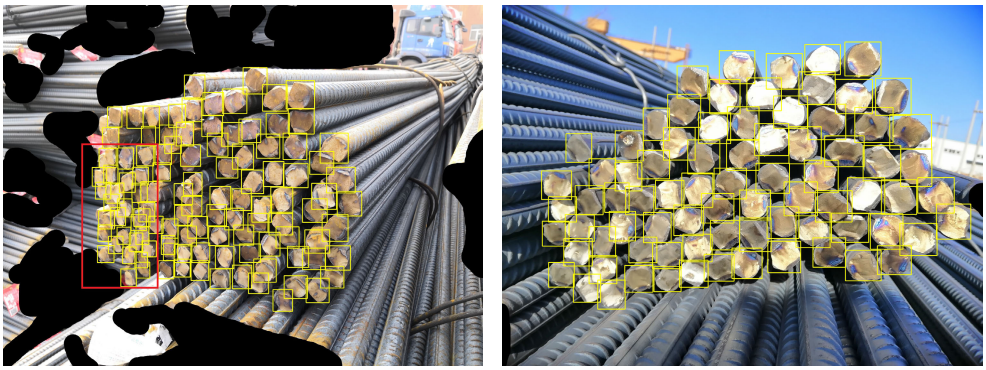
Pixel accuracy and mIOU are used to evaluate the quality of label mask. They are described as (4) and (5) from [32].

$$Pixels\ Accuracy = \frac{\sum_i n_{ii}}{\sum_i T_i} \tag{4}$$

$$mIOU = \frac{1}{n_c} \frac{\sum_i n_{ii}}{\sum_i T_i + \sum_j n_{ji} - n_{ii}} \tag{5}$$

where n_{ij} is the number of pixels of class i that is predicted to belong to class j, where there are n_c different classes, and let$T_i = \sum_j n_{ij}$, T_i is the total number of pixels of class i.

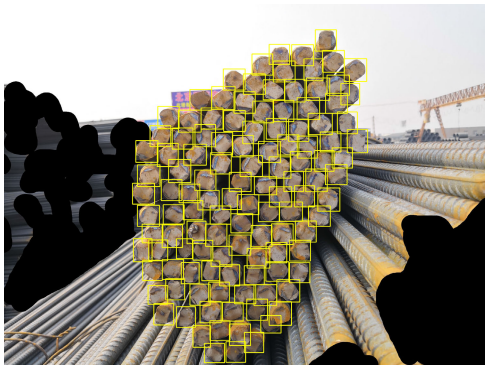
Two training strategies are adopted to train the FCN model, namely, scratch and transfer learning (fine-tuned). In Round6 mentioned in Table 3, six patterns can be achieved through 3 CNN architectures x 2 training strategies. The configuration of each pattern is described as an input size 128×128 pixels, a learning rate 0.01, a weight-decay 0.0001, a max iteration number 1000, batch size 32, learning rate scheduler on miles stone [400,500,600,700,800,900,1000] and gamma 0.8. In FCN model, FLOPS (floating-point oper-



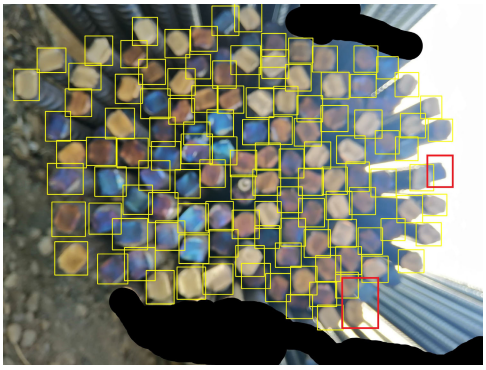
(a) Occlusion

(b) Normal detection 1

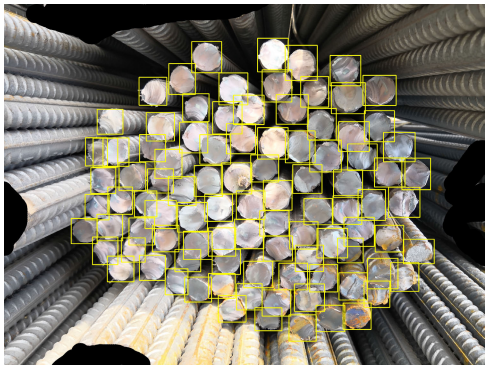
(c) Error detection with wheel



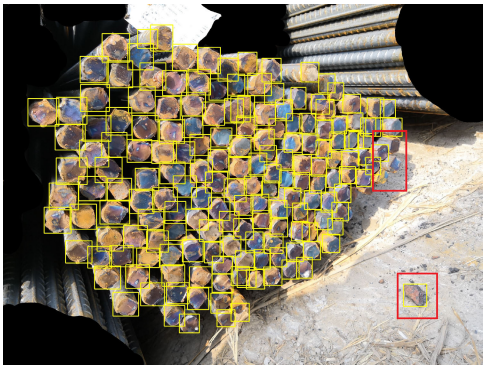
(d) Normal detection 2



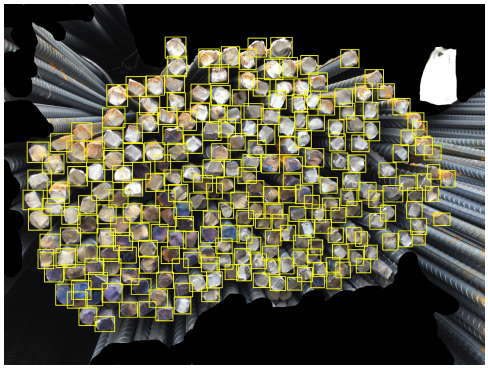
(e) Stronger illumination target loss



(f) Normal detection 3



(g) Error detection with rock



(h) Normal detection 4

FIGURE 13. **Detection result visualization.**

ations per second) are shown in Table 8. ResNet18-FCN has the smallest filter parameter number among the three models of VGG16-FCN, ResNet34-FCN, and ResNet18-FCN. And in the filter parameter number, VGG16-FCN is close to ResNet18-FCN, but the ResNet34-FCN model has twice larger than ResNet18-FCN. For a neural net-

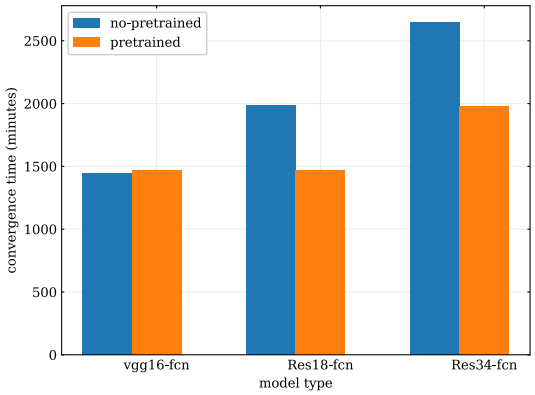


FIGURE 14. Convergence time Round6 between no-pretrained and pretrained.

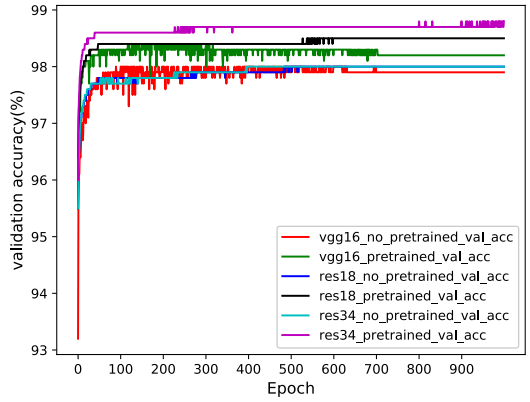


FIGURE 15. Validation accuracy at Round6: VGG16-FCN,ResNet18-FCN,ResNet34-FCN.

work, although fewer filter parameters have less inference time, its performance also depends on FLOPS. By calculating, ResNet18-FCN has 6.1 GFLOPs (Giga FLOPs), and ResNet34-FCN 12.1 GLOPs, VGG16-FCN 55.9 GLOPs, nearly 4 times of ResNet34-FCN model.

Fig.14 shows the convergence time difference between pre-training and no pre-training operations. It demonstrates that except VGG16-FCN, both ResNet34-FCN and ResNet18-FCN have obviously shorter convergence time after pre-training.

Fig.15 shows the validation accuracy comparison of three models (VGG16-FCN, ResNet34-FCN, ResNet18-FCN) with/without pre-training, which proves that the pre-trained models have 0.45% higher validation accuracy in average than non-pre-trained ones. Fig.16 shows the mIOU comparison of three models with/without pre-training, which also proves that the pre-trained models have 1.03% higher mIOU in average than non-pre-trained ones.

Table 9 gives the testing accuracy and mIOU of VGG16-FCN, ResNet18-FCN, and ResNet34-FCN. It demonstrates that ResNet34-FCN with pre-training operation has the highest test accuracy of 98.72% and mIOU of 97.47%.

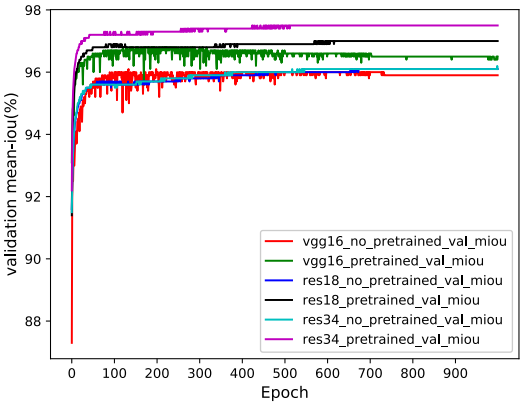


FIGURE 16. Validation mIOU Round6: VGG16-FCN,ResNet18-FCN, ResNet34-FCN.

TABLE 9. VGG16-FCN ResNet18-FCN ResNet34-FCN testset metrics.

Types	test accuracy	test mIOU
VGG16-FCN no pretrained	97.88%	95.84%
VGG16-FCN pretrained	98.19%	96.43%
ResNet18-FCN no pretrained	98.00%	96.08%
ResNet18-FCN pretrained	98.48%	96.99%
ResNet34-FCN no pretrained	98.02%	96.10%
ResNet34-FCN pretrained	**98.72%**	**97.47%**

V. CONCLUSION

The localization and segmentation of the steel bar's end face play a very important role in industrial applications because the traditional manual or MV method is very time-consuming and has low efficiency. Therefore, a new DL detection framework SWDA-CNN is proposed, which contains some new algorithms such as SWDA, Inception-RFB-FPN-based object detection method, FIMLM and modified FCN model. The proposed SWDA can solve the problem of fewer data in small object detection, and Inception-RFB-FPN achieves a trade-off between accuracy and inference time, which means that it can have a high accuracy (Recall: 98.81%, Precision: 97.53%, F1 score: 98.17%) when keeping the least inference time (0.0306s per image). Meanwhile, the proposed FIMLM can overcome the difficulties that it will need massive manpower and spend much time to conduct the labeling work when making the segmentation dataset. At last, the improved pre-trained ResNet34-FCN has an obvious advantage both in convergence time and test accuracy over the non-pre-training FCN model with Test-accuracy of 98.72% and mIOU of 97.47%.

ACKNOWLEDGMENT

The authors would like to thank the anonymous reviewers and the Associate Editor for their valuable comments and suggestions to improve the quality of the manuscript. They would also like to thank Y. Zhu and H. Liu for their valuable suggestions on manuscript grammar and thank C. Tang for his valuable suggestions on manuscript figure drawing and second to fourth authors for their labeling 300 original images and picking up the qualified masks of the rest segmentation dataset.

REFERENCES

[1] L. Sanding, "Design and implementation of k-time-count fault tolerance algorithm," *Comput. Eng. Appl.*, pp. 94–97, Jun. 2004.
[2] Y. Wu, X. Zhou, and Y. Zhang, "Steel bars counting and splitting method based on machine vision," in *Proc. IEEE Int. Conf. Cyber Technol. Autom., Control, Intell. Syst. (CYBER)*, Jun. 2015, pp. 420–425.
[3] D. Zhang, Z. Xie, and C. Wang, "Bar section image enhancement and positioning method in on-line steel bar counting and automatic separating system," in *Proc. Congr. Image Signal Process.*, vol. 2, 2008, pp. 319–323.
[4] X. Ying, X. Wei, Y. Pei-xin, H. Qing-da, and C. Chang-hai, "Research on an automatic counting method for steel bars' image," in *Proc. Int. Conf. Electr. Control Eng.*, Jun. 2010, pp. 1644–1647.
[5] Z. Su, K. Fang, Z. Peng, and Z. Feng, "Rebar automatically counting on the product line," in *Proc. IEEE Int. Conf. Prog. Informat. Comput.*, vol. 2, Dec. 2010, pp. 756–760.
[6] W. Jingzhong, C. Hao, and X. Xiaoqing, "Pattern recognition for counting of bounded bar steel," in *Proc. 4th Int. Conf. Appl. Digit. Inf. Web Technol. (ICADIWT)*, Aug. 2011, pp. 173–176.
[7] Z. Nie, M. H. Hung, and J. Huang, "A novel algorithm of rebar counting on Conveyor Belt based on machine vision," *J. Inf. Hiding Multimed. Sign Process.*, vol. 7, no. 2, pp. 425–437, 2016.
[8] M. F. Ghazali, L.-K. Wong, and J. See, "Automatic detection and counting of circular and rectangular steel bars," in *Proc. 9th Int. Conf. Robotic, Vis., Signal Process. Power Appl.* Singapore: Springer, 2017, pp. 199–207.
[9] X. Yan and X. Chen, "Research on the counting algorithm of bundled steel bars based on the features matching of connected regions," in *Proc. IEEE 3rd Int. Conf. Image, Vis. Comput. (ICIVC)*, Jun. 2018, pp. 11–15.
[10] C. Liu, L. Zhu, and X. Zhang, "Bundled round bars counting based on iteratively trained SVM," in *Proc. International Conference on Intelligent Computing*. Cham, Switzerland: Springer, 2019, pp. 156–165.
[11] Z. Fan, J. Lu, B. Qiu, T. Jiang, K. An, A. N. Josephraj, and C. Wei, "Automated steel bar counting and center localization with convolutional neural networks," 2019, *arXiv:1906.00891*. [Online]. Available: http://arxiv.org/abs/1906.00891
[12] P. Arbelaez, B. Hariharan, C. Gu, S. Gupta, L. Bourdev, and J. Malik, "Semantic segmentation using regions and parts," in *Proc. IEEE Conf. Comput. Vis. Pattern Recognit.*, Jun. 2012, pp. 3378–3385.
[13] M. Kampffmeyer, A.-B. Salberg, and R. Jenssen, "Semantic segmentation of small objects and modeling of uncertainty in urban remote sensing images using deep convolutional neural networks," in *Proc. IEEE Conf. Comput. Vis. Pattern Recognit. Workshops (CVPRW)*, Jun. 2016, pp. 1–9.
[14] H. Noh, S. Hong, and B. Han, "Learning deconvolution network for semantic segmentation," in *Proc. IEEE Int. Conf. Comput. Vis. (ICCV)*, Dec. 2015, pp. 1520–1528.
[15] G. Lin, C. Shen, A. V. D. Hengel, and I. Reid, "Efficient piecewise training of deep structured models for semantic segmentation," in *Proc. IEEE Conf. Comput. Vis. Pattern Recognit. (CVPR)*, Jun. 2016, pp. 3194–3203.
[16] G. Lin, A. Milan, C. Shen, and I. Reid, "RefineNet: Multi-path refinement networks for high-resolution semantic segmentation," in *Proc. IEEE Conf. Comput. Vis. Pattern Recognit. (CVPR)*, Jul. 2017, pp. 1925–1934.
[17] H. Zhao, J. Shi, X. Qi, X. Wang, and J. Jia, "Pyramid scene parsing network," in *Proc. IEEE Conf. Comput. Vis. Pattern Recognit. (CVPR)*, Jul. 2017, pp. 2881–2890.
[18] W. Wang, J. Shen, R. Yang, and F. Porikli, "Saliency-aware video object segmentation," *IEEE Trans. Pattern Anal. Mach. Intell.*, vol. 40, no. 1, pp. 20–33, Jan. 2018.
[19] W. Wang, J. Shen, and L. Shao, "Video salient object detection via fully convolutional networks," *IEEE Trans. Image Process.*, vol. 27, no. 1, pp. 38–49, Jan. 2018.
[20] W. Wang, S. Zhao, J. Shen, S. C. H. Hoi, and A. Borji, "Salient object detection with pyramid attention and salient edges," in *Proc. IEEE/CVF Conf. Comput. Vis. Pattern Recognit. (CVPR)*, Jun. 2019, pp. 1448–1457.
[21] W. Wang, J. Shen, M.-M. Cheng, and L. Shao, "An iterative and cooperative top-down and bottom-up inference network for salient object detection," in *Proc. IEEE/CVF Conf. Comput. Vis. Pattern Recognit. (CVPR)*, Jun. 2019, pp. 5968–5977.
[22] D. Bolya, C. Zhou, F. Xiao, and Y. J. Lee, "YOLACT: Real-time instance segmentation," in *Proc. IEEE/CVF Int. Conf. Comput. Vis. (ICCV)*, Oct. 2019, pp. 9157–9166.
[23] Datafountain.Cn. (2019). *Digital China Innovation Contest, DCIC 2019*. [Online]. Available: https://www.datafountain.cn/competitions/332
[24] M. Kisantal, Z. Wojna, J. Murawski, J. Naruniec, and K. Cho, "Augmentation for small object detection," 2019, *arXiv:1902.07296*. [Online]. Available: http://arxiv.org/abs/1902.07296
[25] L. Castrejon, K. Kundu, R. Urtasun, and S. Fidler, "Annotating object instances with a polygon-RNN," in *Proc. IEEE Conf. Comput. Vis. Pattern Recognit. (CVPR)*, Jul. 2017.
[26] D. Acuna, H. Ling, A. Kar, and S. Fidler, "Efficient interactive annotation of segmentation datasets with Polygon-RNN++," in *Proc. IEEE/CVF Conf. Comput. Vis. Pattern Recognit.*, Jun. 2018.
[27] W. Liu, D. Anguelov, D. Erhan, C. Szegedy, S. Reed, C.-Y. Fu, and A. C. Berg, "SSD: Single shot MultiBox detector," in *Proc. Eur. Conf. Comput. Vis.* Cham, Switzerland: Springer, 2016, pp. 21–37.
[28] S. Ren, K. He, R. Girshick, and J. Sun, "Faster R-CNN: Towards real-time object detection with region proposal networks," in *Proc. Adv. Neural Inf. Process. Syst.*, 2015, pp. 1–9.
[29] K. Chen *et al.*, "MMDetection: Open MMLab detection toolbox and benchmark," 2019, *arXiv:1906.07155*. [Online]. Available: http://arxiv.org/abs/1906.07155
[30] S. Liu, D. Huang, and Y. Wang, "Receptive field block net for accurate and fast object detection," in *Proc. Eur. Conf. Comput. Vis. (ECCV)*, 2018, pp. 404–419.
[31] Github. (2019). *Songwsx/Steel-Detect*. [Online]. Available: https://github.com/songwsx/steel-detect
[32] J. Long, E. Shelhamer, and T. Darrell, "Fully convolutional networks for semantic segmentation," in *Proc. IEEE Conf. Comput. Vis. Pattern Recognit. (CVPR)*, Jun. 2015, pp. 3431–3440.
[33] L. Xiaohu and O. Jineng, "Research on steel bar detection and counting method based on contours," in *Proc. Int. Conf. Electron. Technol. (ICET)*, May 2018, pp. 294–297.

YONGJIAN ZHU received the Ph.D. degree from the Shanghai Institute of Optics and Fine Mechanics, Chinese Academy of Sciences, in 2007. His research interests include machine vision and default detection based on deep learning.

CHULIU TANG received the B.E. degree from the College of Electronic Engineering, Guangxi Normal University, China, in 2017. He is currently pursuing the degree with the College of Electronic Engineering, Guangxi Normal University. His research interests include computer vision, deep learning, and image processing.

HAO LIU received the B.E. degree from the School of Electrical and Electronic Engineering, Wuhan Polytechnic University, Wuhan, China, in 2018. He is currently pursuing the degree with the School of Electronic Engineering, Guangxi Normal University. His research interests are machine vision, default detection, and image processing.

PENGCHI HUANG received the B.E. degree from the Radio and Television Engineering, Xi'an University of Posts and Telecommunications, Xi'an, China, in 2014. He is currently pursuing the degree with the College of Electronic Engineering, Guangxi Normal University. His research interests include computer vision, deep learning, and image processing.