Neurocomputing 410 (2020) 271–283

Contents lists available at ScienceDirect

Neurocomputing

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

A Trajectory-based Attention Model for Sequential Impurity Detection

Wenhao He, Haitao Song, Yue Guo *, Xiaonan Wang, Guibin Bian, Kui Yuan

Institute of Automation, Chinese Academy of Sciences, 95 Zhongguancun East Road, Haidian District, 100190 Beijing, China

ARTICLE INFO

Article history:
Received 2 July 2019
Revised 28 April 2020
Accepted 1 June 2020
Available online 12 June 2020
Communicated by Jungong Han

Keywords:
Impurity detection
Siamese fusion network
Trajectory-based attention model
Sequential region proposal classification

ABSTRACT

Impurity detection involves detecting small impurities in the liquid inside an opaque glass bottle with complex textures by looking through the bottleneck. Sometimes experts have to observe continuous frames to determine the existence of an impurity. In recent years, region-based convolutional neural networks have gained incremental successes in common object detection tasks. However, sequential impurity detections present more challenging issues than detecting targets in a single frame, because consecutive motions and appearance changes of impurities cannot be captured using those common object detectors. In this paper, we propose a simple and controllable ensemble architecture to alleviate this problem. Specifically, a siamese fusion network is used to generate impurity proposals, then an attention model based on visual features and trajectories is proposed to localize a unique region proposal in each frame, finally, a sequential region proposal classifier using a long-term recurrent convolutional network is applied to refine impurity detection performances. The proposed method achieves 79.81% mAP on IML-DET datasets, outperforming a comparable state-of-the-art Mask R-CNN model.

1. Introduction

Impurity detection in a glass container is a process of searching potential impurities [1–6] (color points, fibers, plugs, black points, white points, threads, cotton, and so on) in the liquid by looking through the bottle surface, and containers are usually checked by humans. Specifically, glass bottles can be divided into transparent bottles and opaque bottles: as for transparent bottles, workers can find impurities directly through the bottle walls with their own eyes under strong illuminations, and to adapt to the fast speed of a production line, workers much detect bottles rapidly and constantly, but leak detections are unavoidable after the excessive use of human eyes; as for opaque glass bottles, workers must check them from the bottlenecks, moreover, under limited lighting conditions on bottle surfaces, they have to repeatedly rotate bottles and recheck them from different perspectives.

Impurity detection completely relying on visual features in our task remains challenging due to the following reasons: firstly, a sampled image in an opaque glass not only contains the bottle with complex textures but also is filled with many background fluctuations caused by liquid waves and bubbles after rotating bottles, therefore, traditional machine-vision-based methods [1–4] applied in impurity detection in transparent glass containers with clean backgrounds may be difficult to be applied directly in our sampled images; secondly, ambiguous visual features of impurities in opaque glass containers sometimes lead to challenging discriminations between impurities and backgrounds using state-of-the-art region-based common object detectors [7,8,6].

Trajectory features of impurities in opaque glass bottles are more global and distinctive than local visual features. Specifically, impurity detection in an opaque glass bottle models the problem of searching potential impurities in the liquid inside the opaque glass bottle by looking through the bottleneck. Bottles are usually observed after they are rotated in high speed and then are abruptly stopped [6], as a result, an image sequence observed using our model is shown in Fig. 1, and a circular trajectory can be found if all the locations of an impurity individual are put into a single empty image.

In this paper, a trajectory-based attention model is proposed to alleviate the above-mentioned problems, and a sequential-region-based impurity detection framework (see Fig. 2) is constructed, which eventually improve impurity detection performances. Specifically, to exploit short-term visual changes, a differential image between neighboring frames is combined with a current gray image. To generate high-quality region proposals, a fully convolutional neural network can be applied to produce accurate object proposals [9], inspired by their work, siamese fusion networks are systematically constructed and analyzed to generate high-quality temporal semantic impurity proposals; to eliminate ambiguous background fluctuations for sequential region proposals, a trajectory-based attention model based on circular motion

* Corresponding author.
E-mail address: guoyue2013@ia.ac.cn (Y. Guo).

https://doi.org/10.1016/j.neucom.2020.06.008
0925-2312/

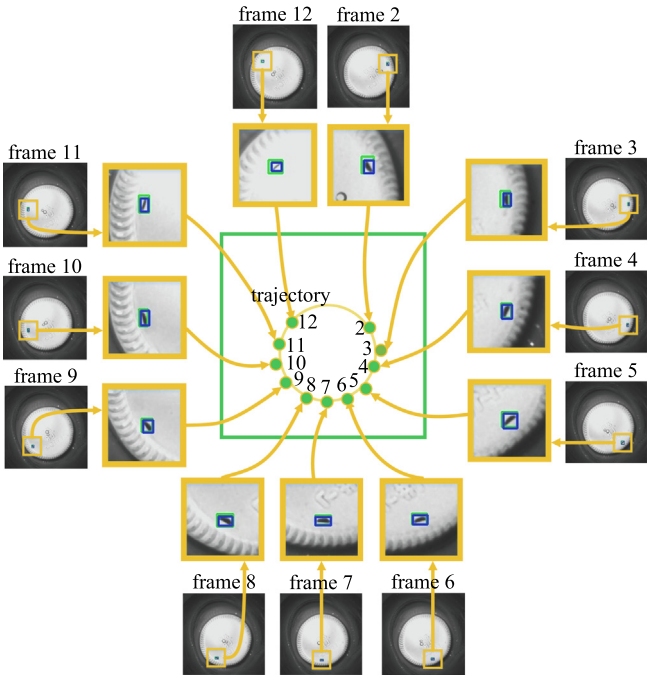


Fig. 1. Impurity detections in an image sequence: the predicted bounding boxes and manually labeled boxes are respectively annotated with blue squares and green rectangles, and a trajectory of the moving impurity is shown in the figure center.

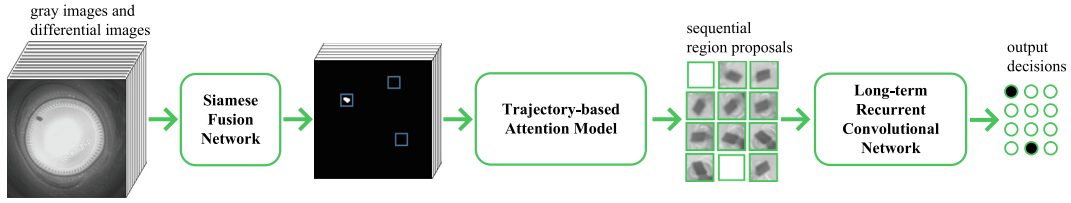


Fig. 2. Sequential-region-based impurity detection framework: in an image sequence, a siamese fusion network takes a gray image and a differential image at each time step as inputs and outputs a binary semantic mask. At each time step, multiple region proposals are cropped from the gray image according to the semantic mask, and their motions are checked globally with a trajectory-based attention model so that all the region proposals with inconsistent motions are eliminated. Finally, the left region proposals at all the time steps are classified by a long-term recurrent convolutional network.

priors is proposed; finally, a long-term recurrent convolutional network is constructed to further ensure consistent visual changes of detected impurities. Quantitative experiments demonstrate that our proposed method outperforms the state-of-the-art region-based object detectors.

The remainder of this paper is organized as follows. Firstly, related definitions and works are introduced in Section 2. Secondly, the sequential-region-based impurity detection framework is described in detail in Section 3. Thirdly, datasets and experiments are elaborated in Section 4. Finally, conclusions, discussions, and future works are presented in Section 5.

The main contributions of this paper are summarized as follows:

(1) A siamese fusion network is proposed to generate temporal semantic impurity proposals, and their variants are systematically constructed and analyzed for fusing multi-domain information.
(2) A trajectory-based attention model is proposed to select the sequential region proposal. Specifically, a small-scale convolutional neural network is trained to further eliminate region proposals of background, and motion priors in an image sequence are integrated to refine the trajectories of impurity proposals.
(3) A long-term recurrent convolutional network is constructed to retain region proposals with consistent appearances in a continuous sequence. To the best of our knowledge, this is the first attempt that simultaneously considers visual appearance changes and global motions in a deep-learning-based impurity detector.
(4) A sequential-region-based impurity detection framework where all the submodels are trained from scratch is built and compared with other state-of-the-art object detectors finetuned in this task, which demonstrates the effectiveness of our method.

2. Related works and definitions

2.1. Related works

2.1.1. Region proposal generation

Region proposals are image patches belonging to the objectness and are directly cropped from static input images. Currently, there are mainly two main-stream approaches to generate region proposals: low-level feature-based generators and high-level-feature-based region generators.

Low-level-feature-based generators output object proposals by grouping small regions into larger ones according to colors, tex-

tures, similarities, and superpixels [6,10]; high-level-feature-based generators provide object proposals with semantic feature information from convolutional neural networks, and they can output object proposals with higher recalls [9].

2.1.2. Region-based object detection

Region-based convolutional neural networks are the most popular deep learning architectures for object detection tasks. At first, thousands of region proposals are generated using selective search, and convolutional neural networks are used for extraction and classification of region features [11]. Then RoI pooling layers are proposed to extract features from regions of interest, while multiple classes and locations of region proposals can be simultaneously determined [12]. Next, region proposal networks are proposed, and shared feature extractions between the region proposals networks and Fast R-CNNs increase the detection speeds [7,13]. Additional target masks can also be fused into an end-to-end object detector to improve performances in some challenging tasks [14]. Moreover, ensembles of convolutional neural networks can be applied to further improve object detection performances. A cascade R-CNN is proposed to detect objects in COCO datasets using the same object detection models with different IoU thresholds [15].

Large numbers of anchor boxes in region-based object detectors lead to a large imbalance between positive and negative region proposals and manual design choices of hyperparameters. To address this problem, on the one hand, an anchor box can be replaced with the top-left corner and the bottom-right corner of a bounding box [16] or four extreme points (top-most, left-most, bottom-most, and right-most) and a center point [17], so anchor box designing and bounding box regression are transferred into keypoint estimation problem; on the other hand, directly predicting confidences for all the object categories and the bounding box on every level of the feature pyramid [18–20] becomes a common solution. Besides, Training balance modules including IoU-balanced sampling, rescaling levels of a feature pyramid, redesigning loss functions [21,22], and generalized IoU [23] are proposed to improve object detection performances.

2.1.3. Salient object detection

Salient object detection is to capture the most attractive object and segment it out from backgrounds in an image [24], which is similar to the region proposal segmentation task.

Salient objects are studied from different perspectives. For example, salient objects are grown with center-surrounding visual attention based on the prior locations of salient objects [25], visually informative patches are extracted with wavelet transform [26], directional cues are represented using quaternionic-distance-based weber descriptor [27], and backgrounds are captured with a local tree-structured low-rank constraint on the representation coefficient matrix [28]. To alleviate data ambiguity and robustly learn in complex scenarios, self-paced learning is integrated to gradually learn from easy training examples to more complex ones [29].

Local contexts are limited with salient object detectors based on convolutional neural networks. To address this issue, a multi-scale cascade network progressively refines detection results from coarse to fine [30], and a multi-scale bidirectional fully convolutional network is built to consolidate multi-level contexts [31]. A two-stream fusion scheme is also effective for fusing multi-level features. For instance, a two-stream part-object assignment network is constructed to reduce noisy assignments from low-level part capsules to high-level object ones [24], and a two-stream fusion scheme is conducted to output fusion maps and the confidence map [32]. Inspired by the above-mentioned work, a siamese fusion network is constructed to capture visual appearances and short-term motions.

2.1.4. Spatial–temporal feature fusion

Long-short term memories (LSTMs) are capable of learning long-term dependencies on public benchmarks and challenging tasks [33]. A long-term recurrent network is connected to the convolutional neural network to form a long-term recurrent convolutional network, and it is trained simultaneously to learn visual representations and temporal relationships in a video sequence [34].

LSTMs can also be combined with convolutional networks [35] using 2D inputs or 3D inputs to capture spatiotemporal information for object tracking [36]. Moreover, temporal pyramid pooling layers are integrated to represent features of videos, then appearance changes and motions are combined to recognize human actions [37]; tubelet proposals are proposed to incorporate temporal and contextual messages for object detections in videos [38]. Since recurrent models using convolutional features in an entire frame may fail to reserve rich dynamics between neighboring frames, an LSTM is integrated with saliency-based 3D-CNN [39].

Similarly, temporal appearance changes of tiny impurities may be imperceptible in a large frame. Inspired by their ideas [37–39], we focus on a specific region in one frame using semantic features, instead of representing a small effective area with feature maps on a whole image.

One of the most similar methods to ours is a facial action unit detection framework encoding region partition rules and integrating a convolutional LSTM to dynamically detect action units [40], different from their work, target regions in our task follow trajectory rules instead of facial partition rules.

2.1.5. Impurity detection

Shape information has been commonly exploited as features of impurities in transparent bottles [1,2], and trajectories of moving blobs are the additional features to separate impurities from bubbles [2]. Region proposals can be classified using machine learning algorithms including support vector machines [3,4]. However, these features and models may be less effective when transferred in opaque glass bottles and evaluated on larger datasets [41].

Convolutional neural networks can be used to simultaneously extract and classify static visual features, but there remain some detection errors [6], furthermore, similarities of two impurity individuals from different frames are studied, but the robustness of impurity detection heavily relies on the augmentation of correlational examples, and the inter-frame correlation is only an implicit cue of local motions [5]. A graph is a flexible architecture to handle complex relations [42]. Transferred to our task, a gallery-guided graph architecture is built to capture inter-sequence relations [43]. However, such an impurity detector relies on a large number of gallery sequences, which may limit its fast adaptation among impurity detection tasks in different domains. To further reduce detection errors using auxiliary cues such as global motions while ignoring a large number of gallery examples during the test stage, we introduce a trajectory extracted from an entire sequence.

2.2. Related definitions

Sequential region proposal: in this paper, a sequential region proposal is selected to maintain the temporal continuity of the region proposals, and there are two main characteristics in it: firstly, a sequential region proposal consists of region proposals that satisfy both the objectness classification rule in the current frame and the motion regularity in an image sequence; secondly, only one region proposal exists in each frame, then they are arranged in the time order as a sequential region proposal.

3. Sequential-region-based impurity detection

3.1. Siamese fusion network

Small-scale models widely used in semantic segmentations are mainly based on fully convolutional networks [44–46] and encoder-decoder architectures [47–51]. To compare the detection performances of our siamese fusion networks, in this paper, all of them are designed using encoders with the same depth.

Training a model from scratch may be difficult to converge if a single gray image is treated as the model input. Compared with backgrounds with complex textures and relatively small motions, rotating impurities are much more visible in differential images, but static impurities are difficult to appear in differential images. Therefore, a gray image and a differential image are constructed as two inputs for every model trained without pre-trained weights, and we respectively extend fully convolutional networks and counterparts with decoders as siamese FE-nets and siamese FD-nets.

Additionally, representative modules used for semantic segmentation can be implemented in our task. Firstly, a convolutional neural network usually includes convolutions, spatial pooling layers, and so on, but objects become less sensitive to location changes using such a network, so boundary details of objects might be lost in the output masks. Based on the above considerations, fully connected conditional random fields [52] can be used to further refine these masks. Secondly, atrous convolutions are typical blocks to avoid oversampling on convolutional feature maps and shrinking the reception fields [45], so we will try to replace spatial poolings with atrous convolutions in the future.

Siamese FE-nets can be constructed by fusing inputs in different layers, but correlations of these inputs become complex when fusing high-level feature maps. To select the most suitable FE-nets for impurity detections, three siamese FE-nets (SFEN-l, SFEN-m, and SFEN–h) are constructed, as shown in Fig. 3.

An FD-net consists of an encoder and a decoder, and a convolutional layer in the encoder can be concatenated with one in the decoder layer to learn the relationships among feature maps in different levels. Similar to the siamese FE-nets, we have designed three siamese FD-nets (SFDN-l, SFDN-m, and SFDN–h), as shown in Fig. 3.

3.2. Trajectory-based attention model

Circular trajectories are formed in image sequences when bottles are rotated and stopped abruptly. Therefore, such motion priors are utilized to design a trajectory-based attention model that consists of two major parts: a region proposal classifier and a trajectory-based attention model.

A convolutional neural network is constructed as a region proposal classifier to classify gray image patches cropped from semantic masks. The model input contains a region proposal which is provided by a siamese fusion network, and its output is a probability of the region proposal belonging to the impurity class. The structures of convolutional layers in this model are the same as those in the sequential region proposal classifier, as shown in Fig. 5.

Given the region proposals in a complete image sequence selected with the above classifier, a motion trajectory is calculated. A specific example is illustrated in Fig. 4, and the process is detailed as below:

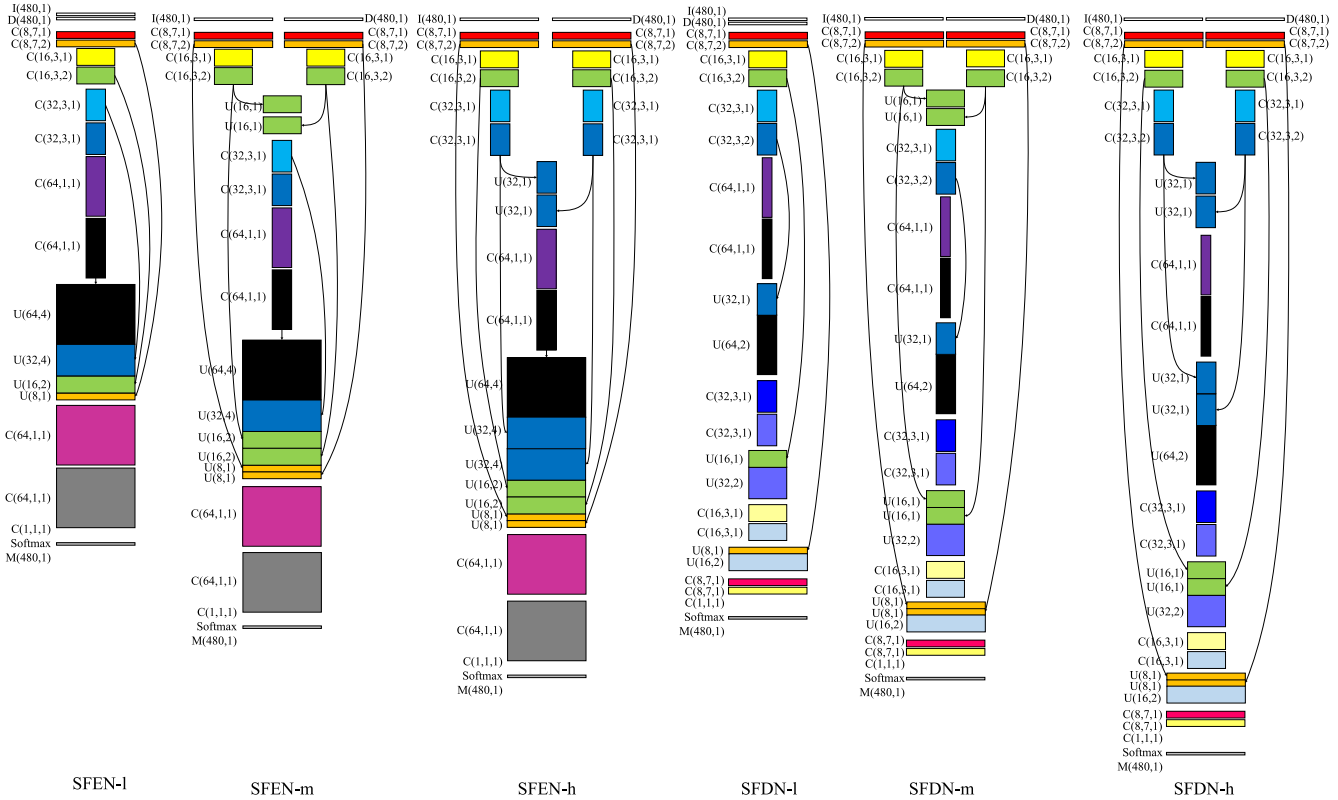


Fig. 3. Architecture of region proposal models: $I(m,n)$ is a $m \times m$ gray image with n channels, similar to the parameters of I, D and M respectively represents a differential image and the corresponding semantic mask; $C(n,k,s)$ is a convolution with n channels, filters in size $k \times k$, and downsampling with s strides; $F(l)$ is a fully-connected layer with l units; $U(n,s)$ is upsampling with n channels and s strides. a gray image and a differential image are organized as inputs of each siamese model, and the network output is a semantic mask. Pixel labels belonging to impurities are labeled as 1s, and background labels are set as 0 s.

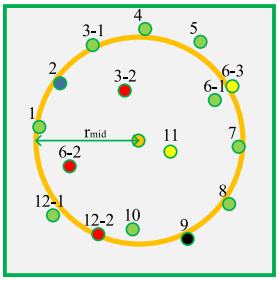


Fig. 4. A trajectory-based attention model based on motion priors: the radius of a circle is r_{mid}; dots with numbers are region proposals in different frames, ij represents that jth region proposal is located at ith frame; states of region proposals are represented with various colors: a black dot means that a true impurity exists, but no region proposals are provided at this location; a green dot or a yellow dot represents the region proposal predicted as an impurity with the classifier, while a blue dot or a red dot is the region proposal predicted as a background.

(1) Given conditions: given the central coordinate of the bottle bottom (x_c, y_c), assume that the ith region proposal in jth frame is p_{i-j}, and the coordinates of its contour center are (x_{i-j}, y_{i-j}), the probability of an impurity from the region proposal classifier is q_+.
(2) Region proposal prediction: region proposals of impurities are selected if predicted probabilities are larger than the probability threshold of impurities q_{t+}, otherwise, those of backgrounds are eliminated.
(3) Distance computation: a L_2 distance from each region proposal left (green and yellow dots) to the bottom center is computed as r_{i-j}, and the mid-value of all the smallest n_{mid} radii is output as r_{mid}, and a circular trajectory is formed using r_{mid}.
(4) Region proposal selection: region proposals which distances from the circle are less than d_{mid} are choosen. In the ith frame, the region proposal with the largest o_+ is selected as a sequential region proposal s_i. For example, when $d_{mid} = 200$, both p_{6-1} and p_{6-3} are close to the circle, but the classification probability of p_{6-1} is larger than that of p_{6-3}, so p_{6-1} should be reserved; p_{11-1} is misclassified by the region proposal classifier but is far away from the circle, so no region proposals can be provided in the 11-th frame.

3.3. Sequential region proposal classification network

Convolutional neural networks and long-short term memories can be concatenated to learn the visual representations and temporal relationships [34], so we construct a long-term recurrent convolutional network to classify image patches at different time steps, as shown in Fig. 5.

In the long-term recurrent convolutional network, the input at each frame is a 40×40 image patch, and weights of convolutional layers in different frames are shared, then each output is a 60 dimensional feature vector. All the vectors can be concatenated and reshaped as the inputs of an LSTM model, and this recurrent model consists of two layers: input of the first layer is a matrix of 12×60, and it outputs a 36 dimensional vector; the second layer takes the output of the first layer as an input and outputs a 36 dimensional vector. Then another fully connected layer with 12 hidden units and a linear activation function are used to predict probabilities of impurities in a specific sequence.

3.4. Independent training of models

Models in our impurity detection framework are Independently trained for multiple tasks including region proposal segmentation, region proposal classification, and sequential region proposal classification.

3.4.1. Region proposal segmentation

The training objective of a siamese fusion network we use is dice coefficient, for the ith sampled image:

$$L_i^{seg} = -\frac{2(m_i^{pred} \cap m_i^{gt})}{m_i^{pred} \cup m_i^{gt}} \tag{1}$$

where m_i^{pred} is a semantic mask predicted with the siamese fusion network, and m_i^{gt} is a corresponding groundtruth pixel map.

3.4.2. Region proposal classification

Regarded as a classification problem of two classes, a small-scale convolutional network outputs probabilities for each region proposal r_i:

$$L_i^{cls} = -o_i^{gt} log(o_i^{pred+}) - (1 - o_i^{gt}) log(o_i^{pred-}) \tag{2}$$

where o_i^{pred+} and o_i^{pred-} respectively represent a probability corresponding to impurity and one belonging to background, and $o_i^{gt} \in \{0, 1\}$ is the groundtruth label.

3.4.3. Sequential region proposal classification

At each time step, an output indicates the probability of a region proposal containing impurities. Empirically, the sequential region proposal classification problem is treated as regression. For ith region q_i in an image patch sequence, the output value of a trained long-term recurrent convolutional network should be close to the ground truth label:

$$L_i^{seq} = \sum_{t=1}^{l_t} || v_{it}^{gt} - v_{it}^{pred} ||_2^2 \tag{3}$$

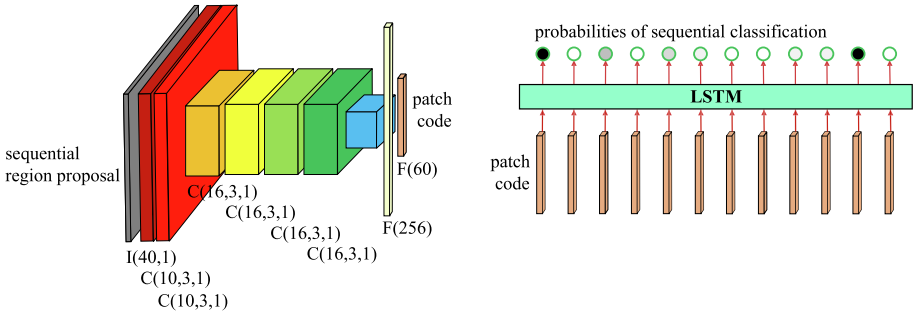


Fig. 5. A long-term recurrent convolutional network to classify region proposals sequentially: the input of this model at each frame is an image patch, and visual features are extracted with a convolutional neural network module, then these features at all the time steps are sent into the long-short-term memory, and probabilities of all the image patches are provided for sequential binary classification.

where at each time step t, v_{it}^{pred} is the predicted value using the network, which should be close to the groundtruth label v_{it}^{gt}, and l_t is the sequence length.

3.4.4. Independent training for multiple tasks

Many researchers have been improving model performances given a fixed dataset and trying to explore the interpretability of their networks. On the contrast, inspired by the idea of AdaBoost [53], a big complex model can be explained with small simple models. Multiple datasets are automatically generated with ground truth labels to adapt for each task, so every small network can be independently trained.

4. Experiments

4.1. Datasets

4.1.1. IML dataset

Images are sequentially sampled after the bottle stops abruptly, and each bottle contains a single impurity. The resolution of each frame is 480×480. To avoid test results affected by model overfitting, different bottles are separately sampled in the training set and the test set. As a result, IML dataset consists of 874 sequences for training and 274 sequences for the test, and IML-DET, IML-SEG, IML-SEQ, and IML-RPN share the same original images from IML dataset. Moreover, the training part and test one for every following dataset are respectively sampled from training sequences and test ones in IML dataset.

4.1.2. IML-DET dataset

Bounding box annotation tools are used to label all the region proposals that have visible impurities. There are 8302 boxes and 2739 ones in the training set and the test set.

4.1.3. IML-SEG dataset

Pixel-level labels are generated with ground truth bounding boxes from IML-DET dataset. Considering that manually labeling pixels takes a much longer time than labeling bounding boxes, pseudo-semantic labels are generated and augmented. Consequently, there are 8302 original images and 33208 augmented ones respectively. Specific augmentation procedures are detailed in Appendix A.

4.1.4. IML-RPN dataset

Region proposals are generated with ground truth bounding boxes in IML-DET dataset. Specifically, after basic data augmentations including translation, rotation, and rescaling, ground truth bounding boxes are used to represent region proposals belonging to impurities. To balance samples of impurities and those of backgrounds, random sampling in a whole frame except regions with impurities is applied to generate additional background region proposals, and the sampling times are 4 in each frame. Different from the small data augmentation scale in IML-SEG dataset, the training set in IML-RPN dataset contains 41568 region proposals with impurities and 40461 ones without impurities, while the test set in IML-RPN dataset includes 13717 region proposals with impurities and 12692 ones without impurities.

4.1.5. IML-SEQ dataset

Sequential region proposals must be generated and augmented because the number of sequences is limited. After sequential region proposal augmentation (see Appendix B), the training set in IML-SEQ dataset contains 199122 region proposals with impurities and 430158 ones without impurities, and the test set in IML-SEQ dataset has 65911 region proposals with impurities and 131369 ones without impurities. Several sequential region proposals are illustrated in Fig. 6 and Fig. 7.

To train the long-term recurrent convolutional network, all the sequential region proposals are derived from the training set. Specifically, sequential region proposals with impurities and those without impurities are mixed, randomly shuffled, and split into training and validation sets with a split ratio of $0.8 : 0.2$.

4.2. Metrics

4.2.1. Overall metrics

Influences of different modules are evaluated in a sequential-region-based impurity detection framework, and at most one ground truth bounding box in a frame is considered in all the evaluations. Assume that a bounding box that contains a true impurity b_g is labeled as l_g, and a predicted bounding box b_p is labeled as l_p.

Specifically, when both b_g and b_p exist, an overlapped ratio is calculated: if b_g and b_p are overlapped, then l_g and l_p are labeled as 1s; if they are not overlapped, then l_g and l_p are respectively 1 and 0; if b_g does not exist but b_p exists, then l_g and l_p will be respectively labeled as 0 and 1; if both b_g and b_p do not exist, then l_g and l_p will be labeled as 0s.

Precisions, recalls, F1 scores, and mAPs are used to evaluate the experimental results [54]. Specifically, precisions, recalls, and F1 scores are computed with macro-average metrics, while mAPs are calculated using micro-average metrics.

4.2.2. Metrics of region proposal segmentation

Region proposals are generated by siamese FE-nets or siamese FD-nets, and semantic region proposals are evaluated using overall metrics.

4.2.3. Metrics of region proposal classification

To determine the class output by the region proposal classifier, decisions have to be made according to the output probabilities of this model. Specifically, a probability of impurity is assumed as q_+, and the corresponding threshold is defined as q_{t+}. If $q_+ > q_{t+}$, then

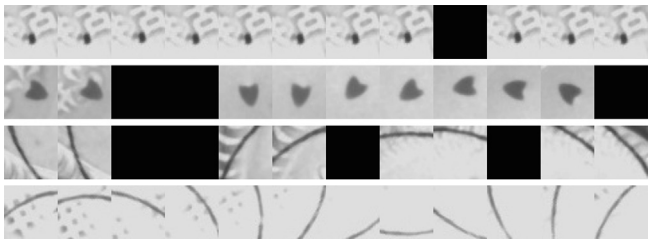


Fig. 6. Sequential region proposals containing impurities: there are four lines of sequences, each line represents a sequential region proposal inside the same bottle, and part of them are randomly blocked.

Fig. 7. Sequential region proposals containing backgrounds, and randomly blocked regions are much more than foreground ones in Fig. 6.

this region proposal may contain an impurity; otherwise, it will be classified as a background.

4.2.4. Metrics of sequential region proposal classification

To obtain the final decision for the sequential region proposal r_{st} at tth time step in the sth sequence, an output o_{st} of the sequential region classification model is transformed to a binary representation. Specifically, given the output threshold o_{st+}, if $o_{st} > o_{st+}$, then r_{st} is classified as an impurity; otherwise, r_{st} is predicted as a background.

4.3. Implementation details

4.3.1. Baselines

Our impurity detection framework is compared with the most famous generic instance segmentation model Mask R-CNN [13] finetuned with IML-DET dataset, and the detection results are evaluated using overall metrics. ResNet backbones [55] and Feature Pyramid Networks [56] are selected to modeling Mask R-CNN [13], base learning rate for training is modified to 0.0002, and the confidence threshold for the test is set as 0.3, then all the other configurations are the default for both training and evaluation.

The trajectory-based attention model is compared with criss-cross attention which is chosen as a baseline of visual attention [57,58] and integrated into siamese fusion networks. Specifically, the criss-cross attention module is integrated between the feature extraction layers and segmentation modules [57], and we have already concatenated earlier feature encoding layers in siamese fusion networks, so in SFEN-l, SFEN-m, and SFEN–h, criss-cross attention modules are added before the feature map concatenation; in SFDN-l, SFDN-m, and SFDN–h, the criss-cross attention modules are added between the encoder and the decoder. For fair comparisons, all the siamese fusion networks with/without criss-cross attention modules are trained from scratch. Besides, it should be noted that this visual attention module and our trajectory-based attention model can be simultaneously applied in an impurity detection framework.

4.3.2. Our method

q_{t+}, o_{st+}, n_{mid}, and d_{mid} are respectively set as $0.5, 0.1, 4$, and 800.

To briefly describe every modules in our framework, siamese fusion network containing an encoder, siamese fusion network including both an encoder and a decoder, criss-cross attention, region proposal classification network, trajectory-based attention model, and sequential region proposal classification network are respectively abbreviated as SFEN, SFDN, CA, CNN, SP, and LRCN.

4.4. Qualitative evaluations

To qualitatively evaluate the sequential impurity detection results, several outputs of SFDN-l + CNN + SP + LRCN and those of the best-performing Mask R-CNN on the same sequences are chosen and compared, as shown in Fig. 8. In Fig. 8, to observe module outputs in our method, unique color is used to annotate boxes output by each module. Specifically, rectangles in violet, brown, red, and orange respectively represent outputs from semantic segmentation models, region classification models, trajectory-based attention models, and long-term recurrent convolutional networks. Besides, orange circles are applied to represent trajectories of detected impurities. It should be noted that different modules are cascaded into a framework, so if an output of a later module exists, so does that of the front modules.

4.5. Quantitative evaluations

4.5.1. Baselines

Publicly available state-of-the-art object detection methods are transferred to our task and evaluated on IML-DET dataset, but the mAP of our proposed method is higher than those of common object detectors, as shown in Table 1.

4.5.2. Siamese fusion networks

Semantic segmentation models including SFEN-l, SFEN-m, SFEN–h, SFDN-l, SFDN-m, and SFDN–h are trained from scratch and evaluated on IML-DET datasets. Specifically, SFDN-l works surprisingly well, and its mAP can reach 80.87% on training data, which is much larger than the second-best model SFEN-l (72.21%), however, 75.64% mAP remained on test data using SFDN-l shows that small overfitting happens on this model. Then mAPs between SFEN-m and SFEN–h are around 48%, but SFDN-m and SFDN–h obtain much lower mAPs. Generally, precisions, recalls, and F1 scores behave similarly with mAPs among all the architectures. Interestingly, the recall of SFDN–h is lower than its precision, but recalls of all the other models are higher than their corresponding precisions, as listed in Table 2.

4.5.3. Region proposal classifier in trajectory-based attention model

The region proposal classifier is evaluated on IML-RPN datasets, and its AUCs evaluated on training data and test one are more than 0.999, probably because sampling rules are the same between training sequences and test counterparts. Specific ROC curves (“cnn - train” and “cnn - test”) are illustrated in Fig. 9.

4.5.4. Long-term reccurent convolutional network

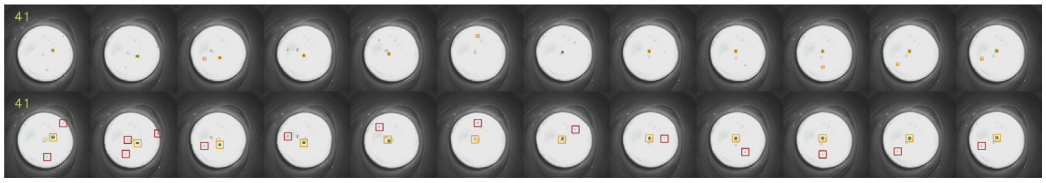
The long-term recurrent convolutional network is evaluated on IML-SEQ datasets, its AUCs remain similarly high with those of the region proposal classifier, and corresponding ROC curves (“lrcn - train” and “lrcn - test”) are detailed in Fig. 9. Datasets for different model functionalities are applied in the region proposal classifier and the long-term recurrent convolutional network, therefore, their classification performances cannot be directly compared.

4.5.5. Ensemble

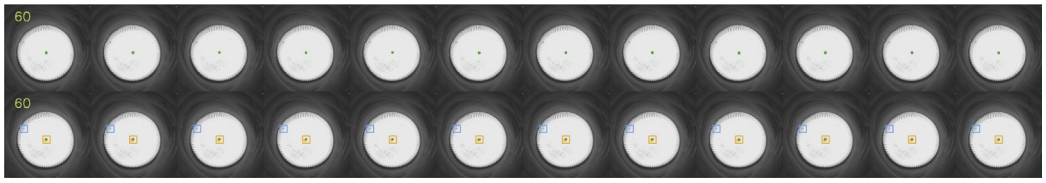
Different modules are concatenated as an ensemble model. The performance of every module can be evaluated in an incremental way, because inputs of a later part may rely on the outputs of front parts. For example, a trajectory-based attention model requires the probabilities of impurity provided by the region proposal classifier, while no probabilities of region proposals can be directly provided by segmentation models; only one region proposal at each frame must be given to long-term recurrent convolutional networks, and only the trajectory-based attention model can output such region proposals in our framework.

By comparing impurity detection results in Table 2, we get the following detailed experimental findings:

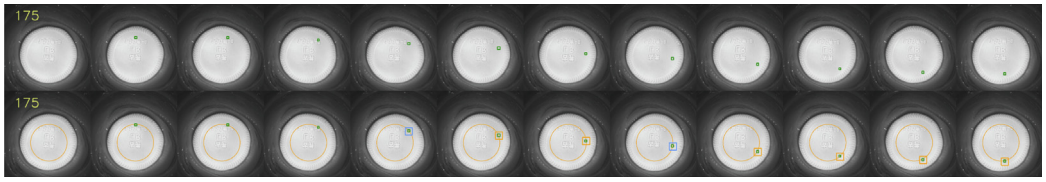
(1) Siamese fusion networks can be used to generate region proposals. For example, the mAP of SFDN-l can achieve 75.64%, which proves the effectiveness of segmentation models applied in impurity detection.
(2) Detection performances of siamese fusion networks with poor results can be improved using the region proposal classifier, but they cannot be better for siamese fusion networks providing high-quality region proposals. All the results including mAPs, precisions, recalls, and F1 scores of SFEN-m, SFEN–h, SFDN-m, and SFDN–h are higher using CNNs. However, they become lower when CNNs are applied to SFEN-l and SFDN-l.



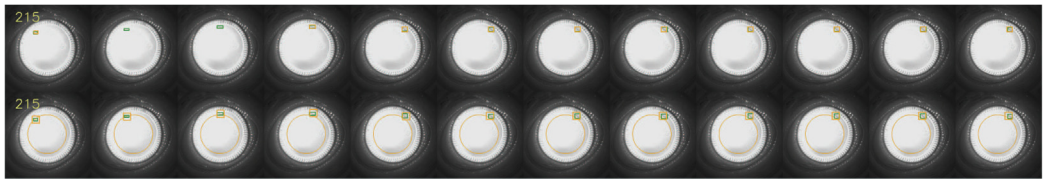
(a) To compare detection outputs using Mask R-CNN (the first row) and the proposed method (the second row), green rectangles represent the ground-truth boxes, and orange ones are the predicted bounding boxes. In sequence 41, small bubbles are classified as impurities using Mask R-CNN in several frames, but the motion trajectory correctly separates impurities and bubbles.



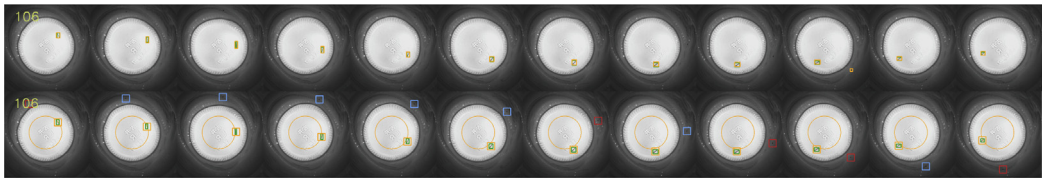
(b) In sequence 60, tiny impurities can be completely detected with our method instead of Mask R-CNN.



(c) Impurity detections in sequence 175 seem more challenging than those in sequence 60, at this time, impurities are again ignored by Mask R-CNN, but some remain visible using our method.



(d) In sequence 215, most of the impurities can be found out using Mask R-CNN, but outputs of our method seem slightly better.



(e) In sequence 106, backgrounds can be mistakenly detected by Mask R-CNN, and sometimes our segmentation model performs much more sensitively. Even in this case, results are still corrected using trajectory-based attention models and long-term recurrent convolutional networks.

Fig. 8. Output comparisons between Mask R-CNN and the sequential impurity detection framework: every test sequence contains 12 frames, and its index is located on its left-top side with orange numbers. To conveniently compare outputs of Mask R-CNN and our method, two sequences with the same sequential index are concatenated vertically, where the top sequence and the bottom one respectively belong to Mask R-CNN and our method.

Table 1
Evaluations of Baselines and Our Method on IML-DET Dataset.

Methods	Backbones	Bottles	mAP	Precision	Recall	F1 score	Bottles	mAP	Precision	Recall	F1 score
Mask R-CNN	R-50-C4	Train	75.81	72.21	83.51	72.13	Test	76.64	70.29	83.58	71.04
Mask R-CNN	R-50-FPN	Train	76.78	74.24	84.73	73.99	Test	77.43	72.65	**86.06**	73.13
Mask R-CNN	R-101-FPN	Train	78.46	75.12	85.70	75.52	Test	77.93	72.92	85.34	73.68
Mask R-CNN	X-101-32x8d-FPN	Train	80.78	76.12	86.61	77.46	Test	78.44	71.79	83.55	73.18
Ours	SFDN-l	Train	**83.90**	**78.75**	**88.23**	**80.78**	Test	**79.81**	**73.21**	84.32	**74.91**

Table 2
Evaluations of Ensembles on IML-DET Dataset.

Models	Bottles	mAP	Precision	Recall	F1 score	Bottles	mAP	Precision	Recall	F1 score
SFEN-l	Train	72.21	68.03	79.81	67.22	Test	68.19	63.78	77.09	61.41
SFEN-l + CNN	Train	67.86	67.71	77.48	64.67	Test	65.81	64.97	76.69	61.21
SFEN-l + CNN + SP	Train	**79.75**	75.14	84.27	76.38	Test	**76.16**	**70.82**	**81.88**	**71.42**
SFEN-l + CNN + SP + LRCN	Train	79.71	**75.22**	**84.49**	**76.41**	Test	75.73	70.69	81.85	71.10
SFEN-m	Train	48.71	56.44	60.77	46.20	Test	49.13	54.45	58.93	44.64
SFEN-m + CNN	Train	51.08	60.22	64.61	49.82	Test	51.60	57.96	63.62	48.70
SFEN-m + CNN + SP	Train	**60.46**	65.26	71.19	**58.94**	Test	**62.38**	63.40	71.37	58.94
SFEN-m + CNN + SP + LRCN	Train	59.94	**66.96**	**72.95**	58.88	Test	62.35	**63.46**	**71.47**	**58.95**
SFEN–h	Train	48.34	54.17	57.53	44.73	Test	47.96	53.44	57.63	42.77
SFEN–h + CNN	Train	52.13	58.11	62.49	49.87	Test	54.08	57.88	64.33	50.08
SFEN–h + CNN + SP	Train	**64.36**	65.95	72.66	**62.08**	Test	68.07	66.49	76.26	64.11
SFEN–h + CNN + SP + LRCN	Train	63.71	**66.96**	**73.87**	61.91	Test	**68.34**	**67.50**	**77.92**	**64.74**
SFDN-l	Train	80.87	75.02	86.46	76.72	Test	75.64	68.58	81.84	69.20
SFDN-l + CNN	Train	71.70	70.74	80.90	68.81	Test	69.14	67.34	79.66	64.78
SFDN-l + CNN + SP	Train	**83.90**	**78.75**	**88.23**	**80.78**	Test	79.81	73.16	84.20	74.88
SFDN-l + CNN + SP + LRCN	Train	82.31	77.82	87.92	79.37	Test	**79.81**	**73.21**	**84.32**	**74.91**
SFDN-m	Train	45.75	50.55	51.04	41.23	Test	48.82	51.27	52.92	41.93
SFDN-m + CNN	Train	50.32	55.42	58.59	47.54	Test	52.91	55.05	59.45	47.87
SFDN-m + CNN + SP	Train	62.14	61.93	67.01	58.99	Test	66.36	61.84	68.44	60.53
SFDN-m + CNN + SP + LRCN	Train	**62.23**	**64.25**	**70.22**	**59.98**	Test	**66.58**	**63.97**	**72.13**	**61.91**
SFDN–h	Train	41.12	48.37	47.17	37.64	Test	41.58	49.16	48.27	37.31
SFDN–h + CNN	Train	45.95	54.22	56.19	44.34	Test	46.93	55.30	59.05	44.38
SFDN–h + CNN + SP	Train	**58.89**	63.62	68.91	57.30	Test	61.53	63.48	71.43	58.39
SFDN–h + CNN + SP + LRCN	Train	58.47	**65.19**	**70.62**	**57.33**	Test	**61.89**	**64.84**	**73.50**	**59.14**

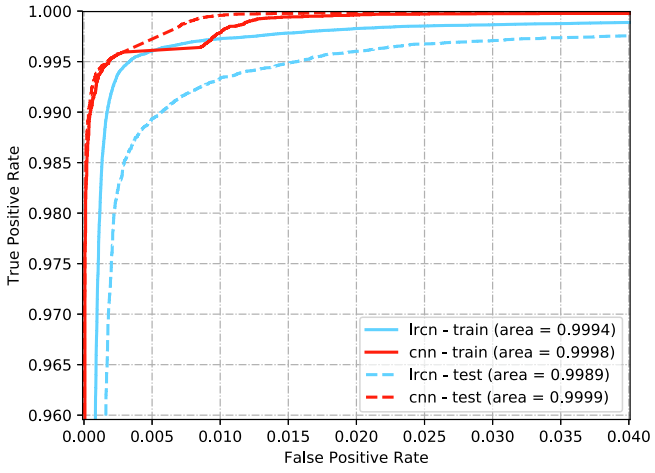


Fig. 9. ROC curves of classification models: "lrcn -train" and "lrcn - test" correspond to a roc curve using the long-term reccurent network evaluated on training data and counterpart predicted on test data, while "cnn - train" and "cnn - test" represent a roc curve using the region proposal classifier computed on training data and that evaluated on test data.

(3) End-to-end object detectors outperforming methods with independently trained sub-modules in single-image detections might be partially explained using our experimental results. Taken our modules as an example, inputs of our region proposal classifier are randomly sampled and may be inconsistent with outputs of siamese fusion networks, but inputs of the object classification network are directly the outputs of the region proposal network in the end-to-end object detection framework. Therefore, some examples trained for our region proposal classifier might never be generated by siamese fusion models, which leads to the inefficient training of our region proposal classifier.

(4) Trajectory-based attention models improve detection performances by a large margin. mAPs of all the different ensembles such as SFEN-l + CNN increase by more than 10% after adding SP modules. Fortunately, relying on trajectory information between consecutive frames in our task, ensembles with independently trained modules even surpass performances of some state-of-the-art end-to-end object detectors. For instance, the mAP of SFDN-l + CNN + SP on test data is 79.81%, and this is higher than those of Mask R-CNNs with 101-layer ResNet backbones.

(5) Long-term recurrent convolutional networks can be used to further refine detection performances. Specifically, precisions and recalls of all the ensembles including LRCNs except the SFEN-l based model are improved, but the increases are much less than those from SPs. For instance, the precision of SFDN-m + CNN + SP increase from 61.84% to 63.97% after adding an LRCN, and its recall climbed by 3.69%. However, in some cases, mAP decreases while all the other metrics increase. For example, mAP of SFEN-l + CNN + SP combined with LRCN drops from 79.75% to 79.71% on training data. Theoretically, continuous appearance changes should be captured using LRCNs, however, sequential region proposals provided by SPs might not be continuous, and training examples provided in IML-SEQ dataset may not be consistent with outputs of SPs. Therefore, slight improvements in practice are obtained with long-term recurrent convolutional networks.

Table 3
Evaluations of Ensembles with Criss-cross Attention on IML-DET Dataset.

Models	Bottles	mAP	Precision	Recall	F1 score	Bottles	mAP	Precision	Recall	F1 score
SFEN-l + CA	Train	61.76	63.06	74.55	57.29	Test	56.58	59.57	71.77	51.05
SFEN-l + CA + CNN	Train	62.98	65.71	75.20	60.15	Test	59.27	62.75	73.71	55.45
SFEN-l + CA + CNN + SP	Train	**77.41**	73.81	83.17	74.28	Test	**75.09**	**70.57**	81.93	**70.95**
SFEN-l + CA + CNN + SP + LRCN	Train	77.36	**74.02**	**83.57**	**74.34**	Test	74.76	70.54	**82.02**	70.43
SFEN-m + CA	Train	11.05	**41.70**	10.39	10.04	Test	12.32	43.93	12.26	11.07
SFEN-m + CA + CNN	Train	14.68	38.65	12.88	13.07	Test	17.25	41.82	15.59	15.04
SFEN-m + CA + CNN + SP	Train	29.49	30.77	22.65	24.83	Test	**35.68**	35.64	27.07	29.02
SFEN-m + CA + CNN + SP + LRCN	Train	**29.59**	30.93	**22.89**	**24.99**	Test	35.58	**36.16**	**27.90**	**29.39**
SFEN–h + CA	Train	16.15	39.20	16.68	14.40	Test	18.59	42.72	20.55	16.27
SFEN–h + CA + CNN	Train	22.37	38.54	23.14	19.81	Test	26.37	43.20	29.36	22.78
SFEN–h + CA + CNN + SP	Train	45.06	41.87	38.56	38.36	Test	51.43	46.35	44.33	42.87
SFEN–h + CA + CNN + SP + LRCN	Train	**45.49**	**43.29**	**40.50**	**39.58**	Test	**51.64**	**47.33**	**45.82**	**43.71**
SFDN-l + CA	Train	80.28	74.43	85.94	76.02	Test	72.56	66.51	79.48	66.04
SFDN-l + CA + CNN	Train	71.76	70.80	80.97	68.88	Test	66.77	65.50	76.63	62.32
SFDN-l + CA + CNN + SP	Train	**84.48**	**79.22**	88.58	**81.36**	Test	**78.64**	**72.28**	83.25	**73.70**
SFDN-l + CA + CNN + SP + LRCN	Train	84.31	79.19	**88.83**	81.27	Test	78.28	72.17	**83.32**	73.44
SFDN-m + CA	Train	42.13	52.73	54.61	40.01	Test	44.26	53.37	56.72	40.83
SFDN-m + CA + CNN	Train	48.05	58.44	61.96	46.95	Test	47.88	57.58	62.58	45.81
SFDN-m + CA + CNN + SP	Train	57.89	64.31	69.49	56.68	Test	59.37	62.72	70.09	56.59
SFDN-m + CA + CNN + SP + LRCN	Train	**58.45**	**65.81**	**71.27**	**57.43**	Test	**59.52**	**63.35**	**71.02**	**56.91**
SFDN–h + CA	Train	10.32	35.07	15.79	9.74	Test	11.47	37.53	15.54	10.65
SFDN–h + CA + CNN	Train	22.97	39.00	31.28	21.78	Test	23.91	40.83	31.70	22.16
SFDN–h + CA + CNN + SP	Train	43.55	46.69	45.34	40.81	Test	44.07	45.55	42.91	39.26
SFDN–h + CA + CNN + SP + LRCN	Train	**44.89**	**49.16**	**48.83**	**42.72**	Test	**45.35**	**48.31**	**47.32**	**41.48**

By comparing impurity detection results in Table 3, we analyze from the following two perspectives:

(1) The criss-cross attention deteriorates the region proposal segmentation model when trained from scratch. For example, mAP of SFDN-l decreases from 75.64% to 72.56% after adding CA. This is probably because the criss-cross attention may be more suitable for re-aggregating complex feature maps from ResNet-50 or ResNet-101 already trained with a large amount of data.
(2) Even in the above-mentioned case, the trajectory-based attention model still improves the overall performances when siamese fusion networks do not perform well. For instance, the mAP of SFDN-l + CA increases up to 78.64% after adding CNN and SP.

5. Conclusion and discussion

In this paper, we propose a sequential framework for impurity detections in opaque glass bottles. Specifically, a siamese FE-net or a siamese FD-net is designed to segment region proposals, then a trajectory-based attention model based on image features and motion priors is proposed to select a sequential region proposal, finally, a long-term recurrent convolutional network is constructed to classify these proposals. Experimental results demonstrate that our framework outperforms the state-of-the-art end-to-end object detector only relying on independent static images.

In the future, we plan to address two major problems about this impurity detection framework: on the one hand, since our trajectory-based attention model is not a general temporal attention model, which limits its applications to other tasks with unpredictable long-term motions, therefore, when motion priors cannot be found, visual attention methods for region proposal segmentation in this task require further studies; on the other hand, to make detectors with the independent training outperform those trained end-to-end, data preprocessing to ensure the consistency of data distributions between two neighboring modules remains tedious, so a more compact automatic sampling scheme during training is necessary for this framework.

CRediT authorship contribution statement

Wenhao He: Supervision. **Haitao Song:** Data curation, Validation. **Yue Guo:** Conceptualization, Methodology, Software, Data curation, Validation, Writing - original draft, Writing - review & editing. **Xiaonan Wang:** Writing - review & editing. **Guibin Bian:** Writing - review & editing. **Kui Yuan:** Supervision.

Acknowledgment

This work is supported by National Key R&D Program of China under Grant 2018YFB1306500, National Key R&D Program of China under Grant 2018YFB1306300, and National Natural Science Foundation (NNSF) of China under Grant 61421004.

Appendix A. Generation and augmentation of pseudo semantic labels

Pixel-wise labeling is far more time consuming than bounding box labeling, so an auxiliary approach is proposed to generate semantic pixels. Bounding box labels can be employed to simultenously update masks and region classes during trainning [59], dense conditional random fields (Dense CRF [52]) have been applied to segment pixel labels with only bounding box annotations [60], and saliency maps can be generated using existing unsupervised salient object detectors [32]. Inspired by the aforementioned works, we generate semantic pixels using Dense CRF. Given bounding boxes B in a $m \times n$ gray image I and a same-sized differential image D, the output mask R is generated as follows:

$$M = \max_{b \in B} \sum_{i=0}^{m} \sum_{j=0}^{n} \sigma(p_{ij} \in b). \tag{4}$$

where b is a bounding box belonging to the set B, p_{ij} is a pixel at the image location (i,j), and σ is an indicator function, so M is a black mask filled with white bounding boxes.

$$IoU(b,M) = \frac{area_b \cap area_M}{area_b \cup area_M}. \tag{5}$$

where $area_o$ is the area inside an object region o, and IoU represents the overlapped ratio between a bounding box b and a refined mask $f_{crf}(M)$ using Dense CRF f_{crf}.

$$w(b,M) = \sigma(IoU(b,f_{crf}(M)) > 0). \tag{6}$$

$$R = \max_{b\in B}\{w(b,M)f_{crf}(M) + (1 - w(b,M))M\}. \tag{7}$$

where $w(b,M)$ indicates whether Dense CRF segments object inside b successfully. If not, the original mask M is used to compensate for this mistake. As a result, the final mask R is obtained as pixel labels for semantic segmentation, and a sample in IML-SEG dataset contains two inputs I and D, and its label is R.

Appendix B. Generation and augmentation of sequential region proposals

An image sequence is taken as an example to illustrate the sampling and augmentation of a sequential region proposal. Augmenting these proposals containing impurities can be seen from Step a) to Step d), and expanding background proposals is described from Step e) to Step f).

(a) Sampling a mask sequence: assume that an image at time step t is I_t , and its groundtruth bounding box is b_t. Then a gray image patch p_t is cropped from the region located using b_t (the center of p_t and that of b_t is coincident, and the size of p_t is 120×120), when no b_t exists in the frame, cropping is skipped. p_t is put into a completely black mask which size is the same as the original image, and a mask at time step t M_t is obtained.

(b) Augmenting a mask sequence: all the masks are rotated in the same sequence simultaneously, and a different angle can be used to rotate each sequence.

(c) Augmenting a sequential region proposal: an augmented mask $M\prime_t$ is taken as an example, contours are found from a mask: if the contour area $s\prime_t > 10$, the minimum enclosing rectangle $b\prime_t$ is reserved, and a gray image patch $p\prime_t$ is cropped at the location of $b\prime_t$ (the center of $p\prime_t$ is the same as that of $b\prime_t$, and its size is 40×40, regions inside the boundary are maintained and resized when $b\prime_t$ meets the image boundary.)

(d) Multiple gray image patches $p\prime_1, p\prime_2, p\prime_3, \ldots p\prime_{T_+}(T_+ < 13)$ make up a positive sequential region proposal including impurities.

(e) A center belonging to a background bounding box q_t is randomly selected in a given image I_t (ranges of horizontal coordinates and vertical counterparts are $(80, 400)$, and their size are all 40×40). IoUs between b_t and q_t are computed: if IoU is larger than 0, then reselect q_t randomly until IoU equals to 0. A gray image patch d_t is then cropped from the region of q_t.

(f) Multiple gray image patches $d\prime_1, d\prime_2, d\prime_3, \ldots d\prime_{T_-}(T_- < 13)$ make up a negative sequential region proposal that include backgrounds.

Positive sequential region proposals generated as the above contain many impurities and a few backgrounds, and all the negative counterparts have backgrounds. In fact, a real sequential region proposal may contain a few impurities or backgrounds or have both of them. Therefore, part of gray image patches should better be blocked. We take a sequential region proposal as an example: suppose the number of blocked region proposals is n_l, n_l time steps are randomly chosen from $[0, 11]$, region proposals at other time steps remain the same, and complete black image patches are used to replace original ones at the blocked time steps. As for a positive sequential region proposal, $n_l \in [0, 4]$, unblocked patches are automatically labeled as 1s, while others are 0s; as for a background region proposal, $n_l \in [0, 11]$, all the image patches are labeled as 0s.

References

[1] B. Zhou, Y. Wang, J. Ge, H. Zhang, A machine vision intelligent inspector for injection, IEEE Pac.-Asia Workshop Comput. Intell. Indust. Appl. (2008) 511–516.
[2] J. Ge, S. Xie, Y. Wang, J. Liu, H. Zhang, B. Zhou, F. Weng, C. Ru, C. Zhou, M. Tan, et al., A system for automated detection of ampoule injection impurities, IEEE Trans. Autom. Sci. Eng. 14 (2) (2017) 1119–1128.
[3] H.J. Liu, A novel vision based inspector with light, Appl. Mech. Mater. (2012) 1916–1921.
[4] B. Huang, S. Ma, Y. Lv, C. Liu, H. Zhang, The study of detecting method for impurity in transparent liquid, Optik 125 (1) (2014) 449–503.
[5] Y. Guo, Y. He, H. Song, W. He, K. Yuan, Correlational examples for convolutional neural networks to detect small impurities, Neurocomputing 295 (21) (2018) 127–141.
[6] Y. Guo, Y. He, H. Song, K. Yuan, Learning to detect small impurities with superpixel proposals, IEEE Int. Conf. Rob. Biomimetics (2017) 320–325.
[7] S. Ren, K. He, R.B. Girshick, J. Sun, Faster r-cnn: towards real-time object detection with region proposal networks, IEEE Trans. Pattern Anal. Mach. Intell. 39 (6) (2017) 1137–1149.
[8] W. Liu, D. Anguelov, D. Erhan, C. Szegedy, S.E. Reed, C. Fu, A.C. Berg, Ssd: Single shot multibox detector, Eur. Conf. Comput. Vision (2016) 21–37.
[9] Z. Jie, W.F. Lu, S. Sakhavi, Y. Wei, E.H. Tay, S. Yan, Object proposal generation with fully convolutional networks, IEEE Trans. Circuits Syst. Video Technol. 28 (1) (2018) 62–75.
[10] J.R.R. Uijlings, K.E.A.V. De Sande, T. Gevers, A.W.M. Smeulders, Selective search for object recognition, Int. J. Comput. Vision 104 (2) (2013) 154–171.
[11] R.B. Girshick, J. Donahue, T. Darrell, J. Malik, Rich feature hierarchies for accurate object detection and semantic segmentation, IEEE Int. Conf. Comput. Vision Pattern Recogn. (2014) 580–587.
[12] R.B. Girshick, Fast r-cnn, IEEE Int. Conf. Comput. Vision (2015) 1440–1448.
[13] K. He, G. Gkioxari, P. Dollar, R.B. Girshick, Mask r-cnn, IEEE Int. Conf. Comput. Vision (2017) 2980–2988.
[14] Y. Mo, G. Han, H. Zhang, X. Xu, W. Qu, Highlight-assisted nighttime vehicle detection using a multi-level fusion network and label hierarchy, Neurocomputing 355 (2019) 13–23.
[15] Z. Cai, N. Vasconcelos, Cascade r-cnn: Delving into high quality object detection, IEEE Int. Conf. Comput. Vision Pattern Recogn. (2018) 6154–6162.
[16] H. Law, J. Deng, Cornernet: detecting objects as paired keypoints, IEEE Eur. Conf. Comput. Vision (2018) 765–781.
[17] X. Zhou, J. Zhuo, P. Krahenbuhl, Bottom-up object detection by grouping extreme and center points, arXiv preprint (2019) arXiv:1901.08043..
[18] T. Kong, F. Sun, H. Liu, Y. Jiang, J. Shi, Foveabox: beyond anchor-based object detector, arXiv preprint (2019) arXiv:1904.03797..
[19] Z. Tian, C. Shen, H. Chen, T. He, Fcos: Fully convolutional one-stage object detection, arXiv preprint (2019) arXiv:1904.01355..
[20] C. Zhu, Y. He, M. Savvides, Feature selective anchor-free module for single-shot object detection, arXiv preprint (2019) arXiv:1903.00621..
[21] J. Pang, K. Chen, J. Shi, H. Feng, W. Ouyang, D. Lin, Libra r-cnn: towards balanced learning for object detection, arXiv preprint (2019) arXiv:1904.02701..
[22] T. Lin, P. Goyal, R.B. Girshick, K. He, P. Dollar, Focal loss for dense object detection, IEEE Int. Conf. Comput. Vision (2017) 2999–3007.
[23] S.H. Rezatofighi, N. Tsoi, J. Gwak, A. Sadeghian, I.D. Reid, S. Savarese, Generalized intersection over union: A metric and a loss for bounding box regression., arXiv preprint (2019) arXiv:1902.09630..
[24] Y. Liu, Q. Zhang, D. Zhang, J. Han, Employing deep part-object relationships for salient object detection, IEEE Int. Conf. Comput. Vision (2019) 1232–1241.
[25] J. Han, K.N. Ngan, M. Li, H. Zhang, Unsupervised extraction of visual attention objects in color images, IEEE Trans. Circuits Syst. Video Technol. 16 (1) (2006) 141–145.
[26] M. Jian, K.-M. Lam, J. Dong, L. Shen, Visual-patch-attention-aware saliency detection, IEEE Trans. Cybern. 45 (8) (2015) 1575–1586.
[27] M. Jian, Q. Qi, J. Dong, Y. Yin, K.-M. Lam, Integrating qdwd with pattern distinctness and local contrast for underwater saliency detection, J. Vis. Commun. Image Represent. 53 (2018) 31–41.
[28] Q. Zhang, Z. Huo, Y. Liu, Y. Pan, C. Shan, J. Han, Salient object detection employing a local tree-structured low-rank representation and foreground consistency, Pattern Recogn. 92 (2019) 119–134.
[29] D. Zhang, D. Meng, J. Han, Co-saliency detection via a self-paced multiple-instance learning framework, IEEE Trans. Pattern Anal. Mach. Intell. 39 (5) (2017) 865–878.

[30] X. Li, F. Yang, H. Cheng, J. Chen, Y. Guo, L. Chen, Multi-scale cascade network for salient object detection, in: ACM International Conference on Multimedia, 2017, pp. 439–447.
[31] F. Yang, X. Li, H. Cheng, Y. Guo, L. Chen, J. Li, Multi-scale bidirectional fcn for object skeleton extraction, in: National Conference on Artificial Intelligence, 2018, pp. 7461–7468.
[32] D. Zhang, J. Han, Y. Zhang, D. Xu, Synthesizing supervision for learning deep saliency network without human annotation, IEEE Trans. Pattern Anal. Mach. Intell. (2020), 1-1.
[33] S. Hochreiter, J. Schmidhuber, Long short-term memory, Neural Comput. 9 (8) (1997) 1735–1780.
[34] J. Donahue, L.A. Hendricks, S. Guadarrama, M. Rohrbach, S. Venugopalan, T. Darrell, K. Saenko, Long-term recurrent convolutional networks for visual recognition and description, IEEE Int. Conf. Comput. Vision Pattern Recogn. (2015) 2625–2634.
[35] J. Redmon, S.K. Divvala, R.B. Girshick, A. Farhadi, You only look once: Unified, real-time object detection, IEEE Int. Conf. Comput. Vision Pattern Recogn. (2016) 779–788.
[36] G. Ning, Z. Zhang, C. Huang, X. Ren, H. Wang, C. Cai, Z. He, Spatially supervised recurrent convolutional neural networks for visual object tracking, IEEE Int. Symp. Circuits Syst. (2017) 1–4.
[37] P. Wang, Y. Cao, C. Shen, L. Liu, H.T. Shen, Temporal pyramid pooling-based convolutional neural network for action recognition, IEEE Trans. Circuits Syst. Video Technol. 27 (12) (2017) 2613–2622.
[38] K. Kang, W. Ouyang, H. Li, X. Wang, Object detection from video tubelets with convolutional neural networks, IEEE Conf. Comput. Vision Pattern Recogn. (2016) 817–825.
[39] X. Wang, L. Gao, J. Song, H.T. Shen, Beyond frame-level cnn: Saliency-aware 3-d cnn with lstm for video action recognition, IEEE Signal Process. Lett. 24 (4) (2017) 510–514.
[40] C. Ma, L. Chen, J. Yong, Au r-cnn: Encoding expert prior knowledge into r-cnn for action unit detection, Neurocomputing 355 (2019) 35–47.
[41] A. Krizhevsky, I. Sutskever, G.E. Hinton, Imagenet classification with deep convolutional neural networks, Adv. Neural Inf. Process. Syst. 141 (5) (2012) 1097–1105.
[42] A. Luo, F. Yang, X. Li, D. Nie, Z. Jiao, S. Zhou, H. Cheng, Hybrid graph neural networks for crowd counting, in: National Conference on Artificial Intelligence, 2020.
[43] W. He, H. Song, Y. Guo, X. Yin, X. Wang, G. Bian, W. Qian, A gallery-guided graph architecture for sequential impurity detection, IEEE Access 7 (2019) 149105–149116.
[44] J. Long, E. Shelhamer, T. Darrell, Fully convolutional networks for semantic segmentation, IEEE Conf. Comput. Vision Pattern Recogn. (2015) 3431–3440.
[45] L. Chen, J.T. Barron, G. Papandreou, K.P. Murphy, A.L. Yuille, Semantic image segmentation with task-specific edge detection using cnns and a discriminatively trained domain transform, IEEE Conf. Comput. Vision Pattern Recogn. (2016) 4545–4554.
[46] L. Chen, G. Papandreou, F. Schroff, H. Adam, Rethinking atrous convolution for semantic image segmentation, arXiv preprint (2017) arXiv:1706.05587..
[47] O. Ronneberger, P. Fischer, T. Brox, U-net: Convolutional networks for biomedical image segmentation, IEEE Int. Conf. Medical Image Comput. Comput.-assisted Intervention (2015) 234–241.
[48] G. Lin, A. Milan, C. Shen, I. Reid, Refinenet: multipath refinement networks with identity mappings for high-resolution semantic segmentation, arXiv preprint (2016) arXiv:1611.06612..
[49] T. Pohlen, A. Hermans, M. Mathias, B. Leibe, Full-resolution residual networks for semantic segmentation in street scenes, IEEE Conf. Comput. Vision Pattern Recogn. (2017) 3309–3318.
[50] A. Islam, M. Rochan, N.D.B. Bruce, Y. Wang, Gated feedback refinement network for dense image labeling, IEEE Conf. Comput. Vision Pattern Recogn. (2017) 4877–4885.
[51] C. Ge, Q. Qu, Y. Gu, Irene, J. Asgeir, Store, Multi-stream multi-scale deep convolutional networks for alzheimer's disease detection using mr images, Neurocomputing 350 (2019) 60–69.
[52] P. Krahenbuhl, V. Koltun, Efficient inference in fully connected crfs with gaussian edge potentials, Adv. Neural Inf. Process. Syst. (2011) 109–117.
[53] L. Breiman, Arcing classifiers, Ann. Stat. 26 (3) (1998) 801–849.
[54] D.M.W. Powers, Evaluation: from precision, recall, and f-factor to roc, informedness, markedness, and correlation, J. Mach. Learn. Technol. 2 (1) (2011) 2229–3981.
[55] K. He, X. Zhang, S. Ren, J. Sun, Deep residual learning for image recognition, IEEE Conf. Comput. Vision Pattern Recogn. (2016) 770–778.
[56] T. Lin, P. Dollar, R.B. Girshick, K. He, B. Hariharan, S.J. Belongie, Feature pyramid networks for object detection, IEEE Conf. Comput. Vision Pattern Recogn. (2017) 936–944.
[57] Z. Huang, X. Wang, L. Huang, C. Huang, Y. Wei, W. Liu, Ccnet: Criss-cross attention for semantic segmentation, IEEE Int. Conf. Comput. Vision (2019) 603–612.
[58] Y. Zhang, K. Li, K. Li, B. Zhong, Y. Fu, Residual non-local attention networks for image restoration, in: International Conference on Learning Representations, 2019..
[59] G. Papandreou, L. Chen, K.P. Murphy, A.L. Yuille, Weakly-and semi-supervised learning of a deep convolutional network for semantic image segmentation, IEEE Int. Conf. Comput. Vision (2015) 1742–1750.
[60] J. Dai, K. He, J. Sun, Boxsup: Exploiting bounding boxes to supervise convolutional networks for semantic segmentation, IEEE Int. Conf. Comput. Vision Pattern Recogn. (2015) 1635–1643.

Wenhao He received the B.E. degree from Beihang University, Beijing, China, and the Ph.D. degree from Institute of Automation, Chinese Academy of Sciences, Beijing, China.
He is currently an Associate Professor at the Institute of Automation, Chinese Academy of Sciences. His research interests involve reconfigurable computing, image processing, and high-performance computing.

Haitao Song received the B.E. degree from Xiamen University, Xiamen, China, in 2009, and the Ph.D. degree in engineering from Institute of Automation, Chinese Academy of Sciences, Beijing, China, in 2014.
He is currently an Assistant Research Fellow with the Institute of Automation, Chinese Academy of Sciences, Beijing, China. His research interests include machine vision and embedded system.

Yue Guo received his B.E. degree in Automation from Lanzhou University of Technology, Gansu, China, in 2013, and the Ph.D. degree from Institute of Automation, Chinese Academy of Sciences, Beijing, China, in 2018.
He is currently postdoctoral with the Institute of Automation, Chinese Academy of Sciences, Beijing, China. His research interests include computer vision, image processing, and machine learning.

Xiaonan Wang received the Bachelor's degree in Mechanical Engineering and Automation from Beijing University of Chemical Technology, Beijing, China, in 2012, and the Doctoral degree in Control Theory and Engineering from Institute of Automation, Chinese Academy of Sciences, Beijing, China, in 2017.
He is an Assistant Research Fellow with the Institute of Automation, Chinese Academy of Sciences, Beijing, China, where his research interests include machine vision and robotic systems.

Guibin Bian received the Bachelor's degree in Mechanical Engineering from North China University of Technology, Beijing, China, in 2004, and the Master's and Doctoral degrees in Mechanical Engineering from Beijing Institute of Technology, Beijing, China, in 2007 and 2010 respectively.

He is a professor at the State Key Laboratory of Management and Control for Complex Systems, Institute of Automation, Chinese Academy of Sciences, Beijing, China, where his research interests include design, sensing, and control for medical robotics.

Kui Yuan received the Master's and Doctoral degrees in Kyushu University, Japan, in 1985 and 1988 respectively.

He is a professor and has been working in Institute of Automation, Chinese Academy of Sciences, Beijing, China. His research interests are in intelligent control, intelligent robots, and machine vision.