AdaptCD: An Adaptive Target Region-based Commodity Detection System

Zeliang Ma[1*], Delong Liu[1*], Zhe Cui[1,2†], Yanyun Zhao[1,2]
[1]Beijing University of Posts and Telecommunications
[2]Beijing Key Laboratory of Network System and Network Culture, China
{mzl,liudelong,cuizhe,zyy}@bupt.edu.cn

Abstract

With the rapid development of computer vision, the detection and counting of goods through computer vision techniques has become practicable. The AICity competition has focused its attention on the automatic recognition and counting of commodities, and has significantly propelled the advancement of this field through the organization of competitive events. Minimizing false positives and false negatives is critical to the success of this task. An ***Adaptive Target Region-based Commodity Detection System*** *has been designed in this study to accurately identify the trajectory and category of goods. To alleviate the difference between training and testing data, two data augmentation methods are utilized, and various data synthesis methods are also designed to meet the training needs of different network models in the framework. Additional Adaptive Algorithms are designed to solve the problem of camera movement during product shooting. An* ***Effective Fusion Algorithm*** *is proposed for Dual Detectors to complement their advantages and minimize detection errors. To maximize the efficiency of the well-trained commodity classifier, an innovative* ***Multi-layer Perception Fusion Module(MPFM)*** *is devised to enhance the commodity classification capabilities, thereby generating more dependable features for tracking purposes. The system has been validated in Multi-Class Product Counting & Recognition for Automated Retail Checkout (2023 AI CITY Challenge Task4) competition, where our results achieve F1 Score of 0.9787 in Task4 testA, ranking second in 2023 AI CITY Challenge Task4 [17]. The code will be released at: https://github.com/mzl163/AICITY23_Task4.*

1. Introduction

In recent years, there has been an increased emphasis in the AI CITY challenge on addressing practical visual problems, which has greatly advanced research in this area [18]. The current year's challenge is centered on multi-object tracking across multiple cameras, natural language-based vehicle retrieval and tracking, recognition of natural driving actions, automatic checkout for multiple products in retail settings, and detection of helmets for motorcycle drivers and passengers. These areas of focus indicate that the 2023 CVPR AI City Challenge Workshop [19] will be dedicated to exploring two promising directions within the field of computer vision — entity retail businesses and intelligent transportation systems.

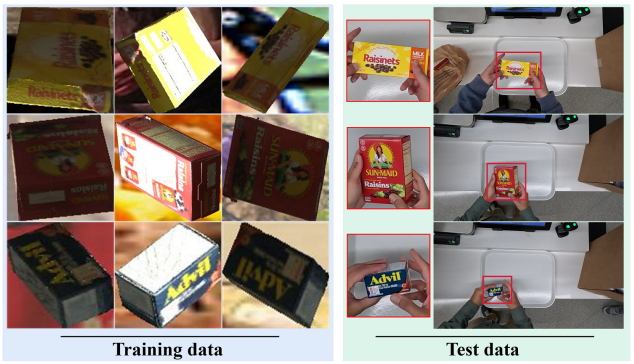

Figure 1. Comparison of Training and Testing Data. The image on the left shows the training set with significant noise and brightness variations, while the frames in the video on the right, which belong to the testing set, have relatively higher quality and more uniform brightness.

The focus of "Multi-Class Product Counting & Recognition for Automated Retail Checkout", Track 4 of the Al City Challenge in 2023, focuses on achieving accurate and automatic check-out in a retail store. This challenge is posed by the real-world challenge of occlusion, movement, similarity among various scanned items, the creation of novel seasonal SKUs, and the cost of misdetection and misclassification [17]. The goal of Task4 track is to achieve accurate detection and tracking of products to avoid counting duplication and to ensure correct classification result. To address this problem, a robust detector is required to

*Equal Contribution
†Corresponding author

accurately locate and track products using a tracking network, following a classification network to determine the product category. Various data augmentation methods and targeted data synthesis approaches are employed to create the training data required for the detector and classifier due to significant differences between the training and testing dataset [15, 35] provided for this competition (Figure 1). Ultimately, satisfactory detection and classification performance are achieved.

Furthermore, the movement of the shooting area in the testing set videos was observed, prompting the proposal of a low-cost target region motion compensation algorithm to eliminate its impact. Object tracking technology plays a crucial role in preventing duplicate counting and accurately locating the product appearance time interval. In our framework, DeepSORT [33] was utilized as the tracker based on the target's features, which resulted in negligible calculation costs. To improve classification and tracking accuracy, MPFM is innovatively designed to fuse feature representations from multiple classifiers to stabilize the classification network's performance, and to provide more reliable product feature representations for tracking. Overall, these techniques and methods lead to satisfactory detection and classification performance in the Al City Challenge 2023.

In light of the above perspectives, a self-adaptive target region-based product detection system is proposed to address the challenges of automated retail checkout. The main contributions of this work are as follows:

1) An **Adaptive Target Area Commodity Detection System** is designed and verified in 2023 AI CITY Challenge Task4, and it achieves very high accuracy in commodity detection.

2) A novel **data augmentation** approach and targeted **data synthesis** methods are developed to create training data that is more representative of real-world scenarios.

3) A **Dual-detector Fusion Algorithm** is designed for the product detection problem, which is used to output the final detection boxes.

4) A cost-effective **Adaptive Target Region Algorithm** is introduced to mitigate the impact of camera movement on object detection and tracking in the test videos.

5) **MPFM** is designed to effectively fuse the feature representations from multiple classifiers and provide reliable product feature representations for tracking, leading to more stable and accurate classification and tracking results.

The proposed system has demonstrated promising performance in accurately detecting and tracking products, and can effectively handle the challenges in automated retail checkout.

2. Related work

In our framework, there are three main components: product detection, product classification, and product tracking. These three components are all critical tasks in computer vision, and there has been a considerable amount of related work. In this section, we summarize some of the work related to our framework.

2.1. Detection

Object detection plays an essential role in the field of computer vision, with the task of detecting the position and category of one or multiple target objects from images or videos. In our framework, the object detection module is a critical component that directly affects the subsequent tracking and classification results. Object detection methods can typically be divided into one-stage [5, 14, 16, 22, 29] and two-stage [4, 7, 13, 21, 23] methods. One-stage methods directly extract features, and then predict target classification and position in the network without region proposals. On the other hand, two-stage methods divide the object detection task into two stages, first generating a series of candidate boxes through a region proposal network, and then performing classification and position regression on these candidate boxes to obtain the final detection results.

2.2. Classification

Image classification is one of the fundamental tasks in computer vision, which aims to classify input images. With the continuous development of deep learning networks, many high-performing classification networks [8, 10, 26, 27] have emerged. Among them, the EfficientNet [28] model adopts a strategy of uniformly scaling depth, width, and resolution, thus improves classification accuracy through a fixed set of scaling factors. The ResNeSt [36] model improves over the ResNet [8] model by stacking up group attention blocks to achieve cross-feature map group properties.

In addition to optimizing the model architecture, data augmentation is also an effective method to improve image classification accuracy. Although training classification networks on large-scale datasets can achieve good results, how to train efficient and accurate classification networks in the absence of sufficient training data remains a challenging problem. Therefore, for small datasets, some smaller and more efficient network architectures (such as MobileNet [9] and ShuffleNet [37]) are also highly regarded.

2.3. Tracking

Multi-object tracking (MOT) aims at tracking multiple moving objects in a video sequence, and is regarded as a

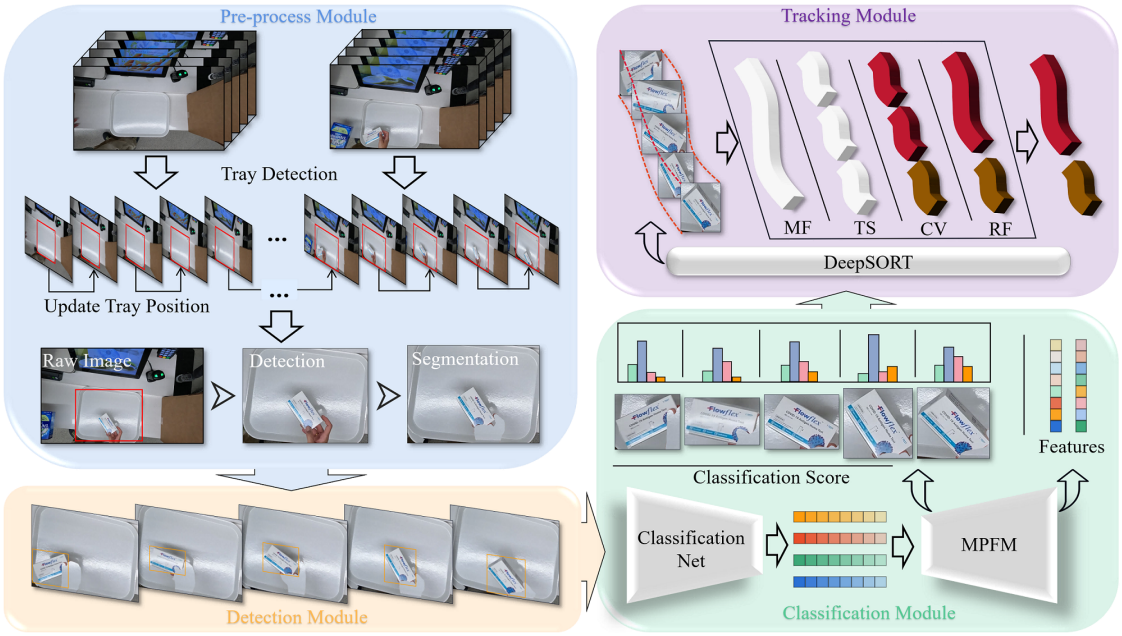

Figure 2. Overview of Adaptive Object Region Detection System Framework. The input of the system is video frames, which are processed by the pre-processing module, detection module, classification module, and tracking module to obtain frames of products entering the tray and their corresponding categories.

research hotspot in the field of computer vision. In recent years, the development of deep learning has facilitated significant advancements in MOT algorithms, leading to successful applications in various practical scenarios. Currently, MOT algorithms can be mainly categorized into two categories: tracking by detection methods methods [1, 34, 38, 39] and feature-based methods [6, 31, 33, 40]. Detection-based methods aim to extract object position and category information through object detection, and then apply associating algorithms for tracking. Feature-based methods, on the other hand, model and match target features (such as shape, color, texture, etc.) to achieve tracking, without relying on detectors. Furthermore, some advanced algorithms also utilize deep learning's feature extraction capabilities to improve tracking accuracy and efficiency.

2.4. Automatic Counting and Recognition of Commodities

In the 2022 AI CITY challenge, the problem of automatic counting and recognition of multi-class commodities received its first attention. During the competition, a batch of high-performing network frameworks has emerged, among which the designated detection and tracking method was the mainstream solution to the automatic counting and recognition of commodities. For example, the work [25] proposed a framework for commodity detection based on target region, which used YOLOv5 [22] as the detector and DeepSORT as the tracker to achieve automated commodity counting. DeepACO [20] used YOLOv4 [3] and human keypoint information to locate the commodity bounding box during the detection stage and captured small objects accordingly. In addition to designing good frameworks, researchers also paid attention to how to make the testing scenes as simple as possible. The work [30] reasonably used an open-source model to accurately locate the region of the commodity to be detected, and found that removing interference from human skin greatly improved the accuracy of commodity detection. Furthermore, VISTA [24] chose a segmentation-based method to replace the detector for commodity localization, extracting the target position through U-Net and tracking the counting. In summary, the efforts of these researchers not only provided effective solutions for the automatic counting and recognition of multi-class commodities, but also laid a solid foundation for subsequent research.

3. Method

The framework proposed in this article consists of four modules: pre-processing module, detection module, classification module, and tracking module, as shown in Figure 2. Each module will be discussed in detail below. The input of the framework is video frames, and after preprocessing, detection, classification and tracking, the final result is obtained.

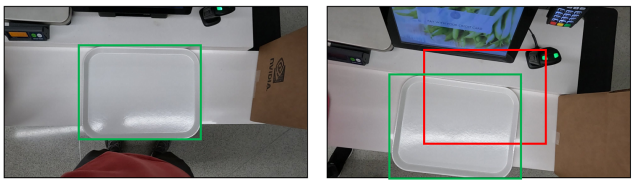

Figure 3. Change of Tray Position: Both images are from TestA Video 3. The left image shows the position of the tray at 0:00, indicated by the green box. The right image shows the position of the tray at 5:02, also indicated by the green box. For comparison, the red box in the right image indicates the position of the tray at 0:00.

3.1. Pre-pocess module

The input video frames need to be first preprocessed before they can undergo subsequent detection and classification operations. To ensure accurate counting result, the tray's position information is extracted using the open-source model DetectoRS to update the target region. To eliminate the background interference during the detection and classification process, the segmentation annotations generated by the open-source model are used to mask the human hands area to eliminate its influence. To achieve this, the pixels in the skin area of human hands are refilled with background pixel value, leading to more precise results (Figure 2 Pre-process Module).

It should be noted that the position of the tray may change in the video (Figure 3), therefore, the method of using a fixed detection area is not reasonable. To address this issue, an algorithm is proposed to adaptively adjust the target detection area with small computational cost. The input parameters for the algorithm include the current frame, the tray region detected in the previous frame, and the current target region, and the output is the updated target region. The algorithm works by first determining whether tray detection is needed in the current frame. When tray detection is needed, the detection result from an open source model is used to calculate the intersection over union (IOU) between the latest detected tray region and the current target region to determine whether the tray has moved. Additionally, the tray region detected in the previous frame is used to determine whether an update is needed, and appropriate processing is performed. To improve computational efficiency, tray detection is performed every 15 frames, and the IOU threshold is used to determine whether the tray has moved. The tray information is updated according to the moving or stopped state.

3.2. Detection Module

This module aims to accurately locate and identify the products appearing in the video, and provide more precise target detection result for subsequent classification. To avoid false detections of the background, the results of two detectors, DetectoRS [21] and Cascade Mask R-CNN [4, 7] are fused. We find that DetectoRS had good detection performance for most targets, but poor detection performance for some small targets and targets obscured by hands or skin. Therefore, we introduced Cascade Mask R-CNN as a supplement.

In the actual implementation, the background influence is eliminated by pre-processing each video frame to fit the objects' actual position. Then, DetectoRS and Cascade Mask R-CNN detectors are used for object detection, which output information such as position, size, and confidence of the targets. To better integrate the advantages of both models, we designed two independent non-maximum suppression algorithms to process the results. Algorithm flow 1 provides specific details. DetectoRS utilized a small-box suppression algorithm, which retained more large boxes and removed small background boxes, while Cascade Mask R-

Algorithm 1 Dual-detector Fusion Algorithm

Input: *result_1*, *result_2*. *result_1* are results of DetectoRS, *result_2* are results of Cascade Mask Rcnn
Output: Fusion result

procedure FUSE_RESULT($result\_1, result\_2$)
  $r\_1 \leftarrow Small\_Box\_Suppression\,(result\_1)$
  $r\_2 \leftarrow Big\_Box\_Suppression\,(result\_2)$
  return $r\_1 \cup r\_2$
end procedure
function SMALL_BOX_SUPPRESSION($bboxes$)
  for $i = 1$ to $bboxes$ **do**
    for $j = i + 1$ to $bboxes$ **do**
      $big\_one, small\_one \leftarrow Compate\,(i, j)$
      $C \leftarrow cross\_area$ **between** i **and** j
      if C/small_area **then**¿thr
        $bboxes \leftarrow bboxes \setminus \{small\_one\}$
      end if
    end for
  end for
end function
function BIG_BOX_SUPPRESSION($bboxes$)
  for $i = 1$ to $bboxes$ **do**
    for $j = i + 1$ to $bboxes$ **do**
      $big\_one, small\_one \leftarrow Compate\,(i, j)$
      $C \leftarrow cross\_area$ **between** i **and** j
      if $C/small\_area > thr$ **then**
        if $small\_area > thr\_area$ **then**
          $bboxes \leftarrow bboxes \setminus \{big\_one\}$
        else
          $bboxes \leftarrow bboxes \setminus \{small\_one\}$
        end if
      end if
    end for
  end for
end function

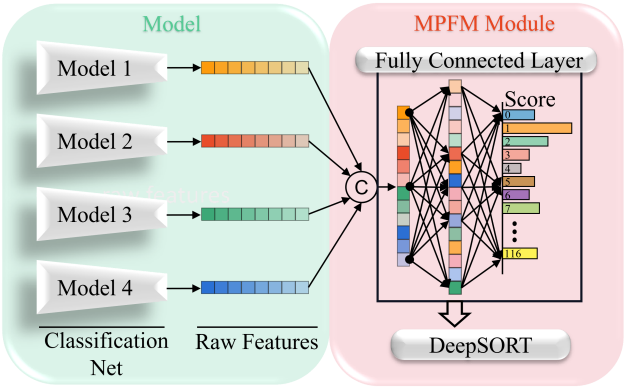

Figure 4. The overall flow of MPFM: The module takes the feature representation of the image from four classification models as input, and sends them to the MPFM for fusion.

CNN used the opposite large-box suppression algorithm to remove large background boxes. Finally, we combined the processed results and sent them to the classifier for classification. This module only focuses on discriminating foreground and background, while specific target categories are considered in the next classification module.

Through the above design and improvements, we have successfully achieved effective detection of targets in videos, and minimized false detections of the background, thus improving the detection accuracy.

3.3. Classification Module

In this module, four CNN-based classification models (EfficientNet-B0, EfficientNet-B2, ResNeSt50, and ResNeSt101) are fused to perform feature extraction and classification. As the detection module only focuses on foreground and background, it is necessary to accurately and stably classify the products in the foreground image in this module. To achieve this goal, we have designed the MPFM (Figure 4) for feature fusion in classification.

To better integrate the features extracted from the four models, the features of each classification model are connected and processed by the MPFM, which is generally a multilayer perceptron (MLP). During training, the four pretrained classification models are frozen, and the image features are obtained by passing the images through these four classifiers. After obtaining the features, they are connected and sent to the MLP layer, where category information is used for supervised learning. This module outputs the features of the MLP intermediate layer and the classification results. The intermediate features will be used for subsequent tracking, while the classification results will be utilized in the post-processing part of the tracking module. This ultimately achieves more accurate classification and more stable tracking.The intermediate features will be used for subsequent tracking, while the classification results will be utilized in the post-processing part of the tracking module. This ultimately achieves more accurate classification and more stable tracking.

Figure 5. Comparison the effects before and after data augmentation. The image on the left is the original image, the image in the middle is after MSRCR data augmentation, and the image on the right is after RetinexNet data augmentation.

3.4. Tracking Module

The tracking module receives the features obtained from the MPFM and conducts DeepSORT-based tracking. Similar to the method proposed in [30], our tracking results are processed by the MTCR module, which splits, classifies, and reconnects the trajectories. In cases where the interval between adjacent frames in the trajectories tracked by DeepSORT is too large, we split them into multiple segments. Next, the frames contained in the segmented trajectories and their corresponding classification results are used for trajectory selection and category determination. Finally, we use the classification results to match and reconnect the retained trajectories. This approach effectively mitigates the impact of detection and classification errors, thereby improving the accuracy and robustness of tracking and obtaining more accurate final results.

4. Experiment

In this section, we will provide a detailed description of our data processing and synthesis methods, as well as explain some experimental details. At the same time, the effectiveness of the proposed method will be demonstrated on the Test A dataset. It should be noted that all testing results were obtained on the Test A dataset.

4.1. Dataset

4.1.1 Data Augmentation

To address the problem of image brightness and jaggedness in the training set, various methods were attempted to reduce the distribution difference between the training and testing sets. To handle overly bright or dark images in the training set, the traditional MSRCR [11] algorithm and deep learning method RetinexNet [12, 32] were utilized. These methods successfully adjusted the brightness level of the training data to match that of the testing set.

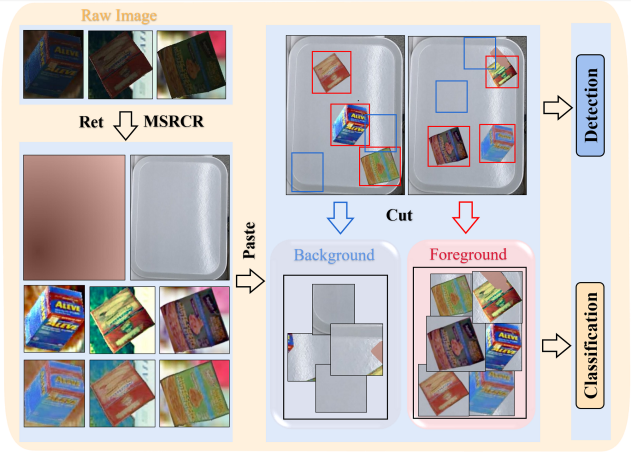

Figure 6. Data Synthesis Method for Detector and Classifier. Training data for the detector was obtained through the upper branch, while training data for the classifier was obtained through the lower branch.

To handle the issue of inconsistent image brightness and jaggedness in the training set, we tested the performance of the two data augmentation methods mentioned above. Although traditional methods like MSRCR are effective in enhancing low-light images, they are time-consuming to process each image and are not suitable for fast batch processing of low-light images. Additionally, images processed with MSRCR may exhibit color deviation issues and differ from the original image. Therefore, we also employed the RetinexNet convolutional neural network model based on the Retinex theory. Compared to traditional methods, this approach is more efficient and has less color deviation between processed images, and the resulting colors tend to be more vivid (Figure 5). To enhance the robustness of the training data, both of these data augmentation methods were used in subsequent data synthesis. Furthermore, the augmented images also underwent Gaussian filtering to reduce the jaggedness problem.

4.1.2 Data Synthesis

Data is essential for training both detectors and classifiers. However, the official dataset provided for the competition only includes product images, which are not sufficient to support the training of good detectors and classifiers. Therefore, it is necessary for us to synthesize the required training data from the original data for both detectors and classifiers (Figure 6).

In the data synthesis process, the issue of inconsistent original brightness of product images was addressed first. for low-light data enhancement, Traditional method MSRCR and deep learning method RetinexNet were employed. By using these methods, we obtain enhanced product images . Next, pre-prepared tray images and human skin simulation images generated by Gaussian mixture model [2] are utilized. Products are randomly pasted into the tray area and covered with human skin randomly to ensure the appropriate coverage ratio of products, and to avoid affecting the training of the detector. Meanwhile, the pasted position information of the product and foreground segmentation labels (required for Cascade Mask R-CNN training) were recorded to obtain training data for the detector. Based on the recorded bounding boxes, foreground images required for training the classifier were cropped (red bounding box). To avoid misclassifying some non-product bounding boxes, a background class was separately designed as training data for the classifier. The training data of the detector were cropped by limiting conditions (blue bounding box), such as only allowing certain corners of the product to appear in the background class, and strictly limiting the ratio. Based on the designed data generation, we have successfully trained detectors and classifiers with high accuracy and robustness.

4.2. Implementation details

In the experiment, an open-source model called DetectoRS was employed as the preprocessing model, and Cascade Mask R-CNN and DetectoRS were utilized as the detection module. In addition, EfficientNet-B0, EfficientNet-B2, ResNeSt50, ResNeSt101, were chosen as the four classification models, and DeepSort was adopted as the tracker to achieve better tracking performance.

During the training phase, a detection model with an input size of 500x400 was used, and it was trained for 5 epochs. The initial learning rate was used for the first 4 epochs, and then it was reduced to 10% of the original value in the fifth epoch. The optimizer used was SGD, and a batch size of 16 was applied during training. Additionally, segmentation labels were added to train the Cascade Mask R-CNN.

In terms of the classifier, the method used in [30] was employed. During the fine-tuning of EfficientNet, the original learning rate was decreased by 90% and only one epoch was trained, while other settings remained unchanged. To better distinguish detected objects from false positives, a background class was added in the classification module.

In the training of the MPFM, the selected four classification models were frozen and trained with the same data. A hidden layer with 2048 neurons and an output layer with 117 neurons were used, and the cross-entropy loss was used as the loss function, with other settings are identical to those of training a single classifier.

4.3. Results

After careful optimization of the framework and models, the performance has reached very good performances, mak-

Rank	Team	F1
1	SKKU Automation Lab	0.9792
2	**BUPT_MCPRL(Ours)**	**0.9787**
3	Zebras	0.8254
4	SCU_Anastasiu	0.8177
5	Fujitsu R&D Center	0.7684
6	Centific	0.6571
7	dtb2023	0.4757
8	Fu	0.4215
9	HCMIU-CVIP	0.3837
10	UTE_AI	0.3441

Table 1. Performance Table of all Teams in TestA.This table shows the performance results of all participating teams in the TestA dataset. Our team name is BUPT_MCPRL and we ranked second in the TestA dataset, with an F1 score only 0.0005 lower than the team ranked first.

ing it more practical in real-world scenarios. It is worth mentioning that our algorithm performs well on the official TestA, with only an almost negligible difference of 0.0005 compared to the best F1 score, while leading more than 0.15 compared with other contestants. The specific TestA results can be seen in Table 1, which lists the leading 10 TestA results.

4.4. Ablation study

To demonstrate the effectiveness of our proposed method, additional analyses are conducted to exame the performance improvement brought by the data augmentation and synthesis methods, the adaptive adjustment detection region algorithm for video frame data preprocessing, the double detector fusion algorithm, and the MPFM.

4.4.1 Data Augmentation and Data Synthesis Methods

Firstly, the effectiveness of data augmentation are investigated. With EfficientNet-b0 used as the classifier, various types of product images, including the original product images and those that had been subjected to different forms of data augmentation, were employed for training. Subsequently, the same detection and tracking methods were used for testing. As shown in Table 2, the first row shows that the classifier's performance was lower when only the original images were used for training.After data augmentation, the test performance improved significantly, and the best performance was achieved by using both data augmentation methods simultaneously. Following the confirmation of the effectiveness of the data augmentation, the effectiveness of the proposed data synthesis method was further explored. The existing method [30] and our data synthesis method were used to generate training data for both detection and classification, and training was conducted accordingly. The

Method	MSRCR	Ret	F1
E-B0	✗	✗	0.7097
	✓	✗	0.8923
	✗	✓	0.7684
	✓	✓	**0.9110**

Table 2. Validation of Data Augmentation: "✓" indicates the use of the MSRCR, and "✗" indicates no use. The same applies to the experiments using RetinexNet(Ret) data augmentation.

Method	MSRCR	Ret	F1
Wans	✓	✓	0.9468
Ours	✓	✓	**0.9787**

Table 3. Validation of Data Synthesis Methods. "✓" indicates the use of the method. "Wans" refers to the data synthesis method described in section 3.1 of [30], and "Ours" refers to the data synthesis method designed in this paper.

final results using the same tracking method are shown in Table 3. Through comparison, it was found that adopting our proposed new data synthesis method can achieve better performance. This demonstrates that the detectors and classifiers trained using skin-occluded products possess stronger generalization ability and can achieve higher test results.

4.4.2 Pre-process Module

This module is proposed to improve the detection of products in the input video frames of the test set. The module enables adaptive adjustment of the detection region, which continuously updates the detection target region. Additionally, an open-source detection model was used to locate the human hand and remove its influence on the skin. To verify the effectiveness of these two improvements, different preprocessing methods were applied to the video frames to generate different preprocessing results, and then the same detection, tracking, and classification methods were used for testing. As shown in Table 4, the worst performance was obtained when no data preprocessing method was used. The use of the adaptive adjustment of the detection region algorithm or the human hand skin removal method alone can improve the testing performance. And the best testing performance was obtained when both methods were used simultaneously, demonstrating the effectiveness of the proposed methods.

4.4.3 Dual Detector Fusion Algorithm

To avoid detecting background erroneously, two object detectors, DetectoRS and Cascade Mask R-CNN, were introduced. In this study, the results obtained from using DetectoRS and Cascade Mask R-CNN individually as well as

Skin Mask	ATRA	F1
✗	✗	0.7097
✗	✓	0.7187
✓	✗	0.9565
✓	✓	**0.9787**

Table 4. Validation of the Pre-process Module. "✓" in the table indicates the use of the method, "✗" indicates the non-use. "Skin Mask" represents the method of using an open-source model to remove the influence of skin, and "ATRA" represents the Adaptive Target Region Algorithm.

DetectoRS	Cascade Mask R-CNN	Method	F1
✓	✗	✗	0.9528
✗	✓	✗	0.9215
✓	✓	✗	0.9565
✓	✓	NMS	0.9468
✓	✓	**Ours**	**0.9787**

Table 5. Validation of the Dual Detector Fusion Algorithm. "✓" indicates the use of the method, "✗" indicates not. "NMS" refers to the conventional non-maximum suppression algorithm used for fusing dual detectors, and "Ours" refers to the fusion method proposed in this paper.

the fusion results with different fusion methods were used as the detection results, and the same classification and tracking methods were used for testing. As shown in Table 5, the first two rows of the table represent the test results obtained from using a single detector that DetectoRS performing better. The last three rows of the table represent the results obtained by merging the results of the two detectors. The results show that the performance is lower without using any fusion method, possibly because simply merging the results retains more background false positives. The performance of the NMS fusion is even worse than that of DetectoRS alone, possibly due to some high-confidence false detections of Cascade Mask R-CNN. However, our proposed dual-detector fusion algorithm can achieve the best test results, demonstrating the simplicity and effectiveness of our method for commodity detection.

4.4.4 MPFM

To better utilize the features extracted by four classification models, the MPFM is proposed. In this section, the features output by a single model and the features obtained from different feature fusion methods are sent to the tracker, and the testing uses the same detection method. Table 6 presents the test results of various methods. Among them, the DTC method refers to the method proposed in [30]. The AdaptPooling method refers to pooling the features output by multiple classification models before sending them to the tracker. The results show that the test results of a single model are lower than those of model fusion. Furthermore, using the MPFM for classifier feature fusion can achieve the best test results, demonstrating the effectiveness of our proposed model fusion method.

Method	F1
EfficientNet-B0	0.9110
EfficientNet-B2	0.9110
ResNeSt50	0.9215
ResNeSt101	0.8923
DTC	0.9680
AdaptPooling	0.9326
MPFM (Ours)	**0.9787**

Table 6. MPFM Validation. "EfficientNet-B0", "EfficientNet-B2", "ResNeSt50", and "ResNeSt101" respectively indicate the use of individual classification models for classification and the output features are sent to DeepSORT.

5. Conclusion

The practical application of visual technology for product detection and counting has become increasingly prevalent with the development of computer vision. In this study, we proposed a adaptive object detection system that can accurately identify the trajectory and category of products, and can achieve very good performance in the 2023 AI CITY Challenge Task4 competition which greatly reduces false detections and missed detections. To address the differences between training and testing data, various synthetic data approaches have been specially designed to meet the training requirements of different network models. An adaptive adjustment algorithm for region detection has been additionally developed to address the problem of camera movement when capturing products. Also, a dual detector fusion algorithm has been used to minimize false detections. In the last, MPFM has been innovatively designed to further improve the product's classification ability to output more reliable features for tracking. Detailed ablation experiments have been conducted for each adapted improvement to demonstrate its effectiveness. However, our system does not meet real-time requirements (Single NVIDIA GeForce RTX 3090 achieves 3 FPS). In the future, greater attention will be paid to balancing the accuracy and efficiency of the method, to ensure the meaningfulness of the proposed approach.

6. Acknowledgements

This work is supported by Chinese National Science Foundation under Grants 62206026.

References

[1] Alex Bewley, Zongyuan Ge, Lionel Ott, Fabio Ramos, and Ben Upcroft. Simple online and realtime tracking. In *2016 IEEE international conference on image processing (ICIP)*, pages 3464–3468. IEEE, 2016. 3

[2] Christopher M Bishop. Mixture density networks. 1994. 6

[3] Alexey Bochkovskiy, Chien-Yao Wang, and Hong-Yuan Mark Liao. Yolov4: Optimal speed and accuracy of object detection. *arXiv preprint arXiv:2004.10934*, 2020. 3

[4] Zhaowei Cai and Nuno Vasconcelos. Cascade r-cnn: Delving into high quality object detection. In *Proceedings of the IEEE conference on computer vision and pattern recognition*, pages 6154–6162, 2018. 2, 4

[5] Nicolas Carion, Francisco Massa, Gabriel Synnaeve, Nicolas Usunier, Alexander Kirillov, and Sergey Zagoruyko. End-to-end object detection with transformers. In *Computer Vision–ECCV 2020: 16th European Conference, Glasgow, UK, August 23–28, 2020, Proceedings, Part I 16*, pages 213–229. Springer, 2020. 2

[6] Long Chen, Haizhou Ai, Zijie Zhuang, and Chong Shang. Real-time multiple people tracking with deeply learned candidate selection and person re-identification. In *2018 IEEE international conference on multimedia and expo (ICME)*, pages 1–6. IEEE, 2018. 3

[7] Kaiming He, Georgia Gkioxari, Piotr Dollár, and Ross Girshick. Mask r-cnn. In *Proceedings of the IEEE international conference on computer vision*, pages 2961–2969, 2017. 2, 4

[8] Kaiming He, Xiangyu Zhang, Shaoqing Ren, and Jian Sun. Deep residual learning for image recognition. In *Proceedings of the IEEE conference on computer vision and pattern recognition*, pages 770–778, 2016. 2

[9] Andrew G Howard, Menglong Zhu, Bo Chen, Dmitry Kalenichenko, Weijun Wang, Tobias Weyand, Marco Andreetto, and Hartwig Adam. Mobilenets: Efficient convolutional neural networks for mobile vision applications. *arXiv preprint arXiv:1704.04861*, 2017. 2

[10] Forrest N Iandola, Song Han, Matthew W Moskewicz, Khalid Ashraf, William J Dally, and Kurt Keutzer. Squeezenet: Alexnet-level accuracy with 50x fewer parameters and¡ 0.5 mb model size. *arXiv preprint arXiv:1602.07360*, 2016. 2

[11] Daniel J Jobson, Zia-ur Rahman, and Glenn A Woodell. A multiscale retinex for bridging the gap between color images and the human observation of scenes. *IEEE Transactions on Image processing*, 6(7):965–976, 1997. 5

[12] Edwin H Land and John J McCann. Lightness and retinex theory. *Josa*, 61(1):1–11, 1971. 5

[13] Tsung-Yi Lin, Piotr Dollár, Ross Girshick, Kaiming He, Bharath Hariharan, and Serge Belongie. Feature pyramid networks for object detection. In *Proceedings of the IEEE conference on computer vision and pattern recognition*, pages 2117–2125, 2017. 2

[14] Tsung-Yi Lin, Priya Goyal, Ross Girshick, Kaiming He, and Piotr Dollár. Focal loss for dense object detection. In *Proceedings of the IEEE international conference on computer vision*, pages 2980–2988, 2017. 2

[15] Tsung-Yi Lin, Michael Maire, Serge Belongie, James Hays, Pietro Perona, Deva Ramanan, Piotr Dollár, and C Lawrence Zitnick. Microsoft coco: Common objects in context. In *Computer Vision–ECCV 2014: 13th European Conference, Zurich, Switzerland, September 6-12, 2014, Proceedings, Part V 13*, pages 740–755. Springer, 2014. 2

[16] Wei Liu, Dragomir Anguelov, Dumitru Erhan, Christian Szegedy, Scott Reed, Cheng-Yang Fu, and Alexander C Berg. Ssd: Single shot multibox detector. In *Computer Vision–ECCV 2016: 14th European Conference, Amsterdam, The Netherlands, October 11–14, 2016, Proceedings, Part I 14*, pages 21–37. Springer, 2016. 2

[17] Milind Naphade and Rama et al. Chellappa. Ai city challenge, 2023. https://www.aicitychallenge.org/. 1

[18] M. Naphade, S. Wang, D. C. Anastasiu, Z. Tang, M. Chang, Y. Yao, L. Zheng, M. Shaiqur Rahman, A. Venkatachalapathy, A. Sharma, Q. Feng, V. Ablavsky, S. Sclaroff, P. Chakraborty, A. Li, S. Li, and R. Chellappa. The 6th ai city challenge. In *2022 IEEE/CVF Conference on Computer Vision and Pattern Recognition Workshops (CVPRW)*, pages 3346–3355. IEEE Computer Society, June 2022. 1

[19] Milind Naphade, Shuo Wang, David C. Anastasiu, Zheng Tang, Ming-Ching Chang, Yue Yao, Liang Zheng, Mohammed Shaiqur Rahman, Meenakshi S. Arya, Anuj Sharma, Qi Feng, Vitaly Ablavsky, Stan Sclaroff, Pranamesh Chakraborty, Sanjita Prajapati, Alice Li, Shangru Li, Krishna Kunadharaju, Shenxin Jiang, and Rama Chellappa. The 7th AI City Challenge. In *The IEEE Conference on Computer Vision and Pattern Recognition (CVPR) Workshops*, June 2023. 1

[20] Long Hoang Pham, Duong Nguyen-Ngoc Tran, Huy-Hung Nguyen, Tai Huu-Phuong Tran, Hyung-Joon Jeon, Hyung-Min Jeon, and Jae Wook Jeon. Deepaco: A robust deep learning-based automatic checkout system. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, pages 3107–3114, 2022. 3

[21] Siyuan Qiao, Liang-Chieh Chen, and Alan Yuille. Detectors: Detecting objects with recursive feature pyramid and switchable atrous convolution. In *Proceedings of the IEEE/CVF conference on computer vision and pattern recognition*, pages 10213–10224, 2021. 2, 4

[22] Joseph Redmon, Santosh Divvala, Ross Girshick, and Ali Farhadi. You only look once: Unified, real-time object detection. In *Proceedings of the IEEE conference on computer vision and pattern recognition*, pages 779–788, 2016. 2, 3

[23] Shaoqing Ren, Kaiming He, Ross Girshick, and Jian Sun. Faster r-cnn: Towards real-time object detection with region proposal networks. *Advances in neural information processing systems*, 28, 2015. 2

[24] Md Shihab, Istiak Hossain, Nazia Tasnim, Hasib Zunair, Labiba Kanij Rupty, and Nabeel Mohammed. Vista: Vision transformer enhanced by u-net and image colorfulness frame filtration for automatic retail checkout. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, pages 3183–3191, 2022. 3

[25] Maged Shoman, Armstrong Aboah, Alex Morehead, Ye Duan, Abdulateef Daud, and Yaw Adu-Gyamfi. A region-

based deep learning approach to automated retail checkout. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, pages 3210–3215, 2022. 3

[26] Christian Szegedy, Sergey Ioffe, Vincent Vanhoucke, and Alexander Alemi. Inception-v4, inception-resnet and the impact of residual connections on learning. In *Proceedings of the AAAI conference on artificial intelligence*, volume 31, 2017. 2

[27] Christian Szegedy, Wei Liu, Yangqing Jia, Pierre Sermanet, Scott Reed, Dragomir Anguelov, Dumitru Erhan, Vincent Vanhoucke, and Andrew Rabinovich. Going deeper with convolutions. In *Proceedings of the IEEE conference on computer vision and pattern recognition*, pages 1–9, 2015. 2

[28] Mingxing Tan and Quoc Le. Efficientnet: Rethinking model scaling for convolutional neural networks. In *International conference on machine learning*, pages 6105–6114. PMLR, 2019. 2

[29] Mingxing Tan, Ruoming Pang, and Quoc V Le. Efficientdet: Scalable and efficient object detection. In *Proceedings of the IEEE/CVF conference on computer vision and pattern recognition*, pages 10781–10790, 2020. 2

[30] Junfeng Wan, Shuhao Qian, Zihan Tian, and Yanyun Zhao. An effective framework of multi-class product counting and recognition for automated retail checkout. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, pages 3282–3290, 2022. 3, 5, 6, 7, 8

[31] Zhongdao Wang, Liang Zheng, Yixuan Liu, Yali Li, and Shengjin Wang. Towards real-time multi-object tracking. In *Computer Vision–ECCV 2020: 16th European Conference, Glasgow, UK, August 23–28, 2020, Proceedings, Part XI 16*, pages 107–122. Springer, 2020. 3

[32] Chen Wei, Wenjing Wang, Wenhan Yang, and Jiaying Liu. Deep retinex decomposition for low-light enhancement. *arXiv preprint arXiv:1808.04560*, 2018. 5

[33] Nicolai Wojke, Alex Bewley, and Dietrich Paulus. Simple online and realtime tracking with a deep association metric. In *2017 IEEE international conference on image processing (ICIP)*, pages 3645–3649. IEEE, 2017. 2, 3

[34] Jialian Wu, Jiale Cao, Liangchen Song, Yu Wang, Ming Yang, and Junsong Yuan. Track to detect and segment: An online multi-object tracker. In *Proceedings of the IEEE/CVF conference on computer vision and pattern recognition*, pages 12352–12361, 2021. 3

[35] Yue Yao, Liang Zheng, Xiaodong Yang, Milind Napthade, and Tom Gedeon. Attribute descent: Simulating object-centric datasets on the content level and beyond. *arXiv preprint arXiv:2202.14034*, 2022. 2

[36] Hang Zhang, Chongruo Wu, Zhongyue Zhang, Yi Zhu, Haibin Lin, Zhi Zhang, Yue Sun, Tong He, Jonas Mueller, R Manmatha, et al. Resnest: Split-attention networks. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, pages 2736–2746, 2022. 2

[37] Xiangyu Zhang, Xinyu Zhou, Mengxiao Lin, and Jian Sun. Shufflenet: An extremely efficient convolutional neural network for mobile devices. In *Proceedings of the IEEE conference on computer vision and pattern recognition*, pages 6848–6856, 2018. 2

[38] Yifu Zhang, Chunyu Wang, Xinggang Wang, Wenjun Zeng, and Wenyu Liu. Fairmot: On the fairness of detection and re-identification in multiple object tracking. *International Journal of Computer Vision*, 129:3069–3087, 2021. 3

[39] Xingyi Zhou, Vladlen Koltun, and Philipp Krähenbühl. Tracking objects as points. In *Computer Vision–ECCV 2020: 16th European Conference, Glasgow, UK, August 23–28, 2020, Proceedings, Part IV*, pages 474–490. Springer, 2020. 3

[40] Ji Zhu, Hua Yang, Nian Liu, Minyoung Kim, Wenjun Zhang, and Ming-Hsuan Yang. Online multi-object tracking with dual matching attention networks. In *Proceedings of the European conference on computer vision (ECCV)*, pages 366–382, 2018. 3