electronics

MDPI

Article

Design of Efficient Human Head Statistics System in the Large-Angle Overlooking Scene

An Wang [1], Xiaohong Cao [1], Lei Lu [1], Xinjing Zhou [2,*] and Xuecheng Sun [3,*]

[1] Shanghai Feilo Acoustics Co. Ltd., Shanghai 200233, China; wangan@yaming-lighting.com (A.W.); red.cao@yaming-lighting.com (X.C.); lulei@yaming-lighting.com (L.L.)
[2] Microelectronic Research & Development Center, Shanghai University, Shanghai 200444, China
[3] Department of Mechatronics Engineering and Automation, Shanghai University, Shanghai 200444, China
* Correspondence: zhouxinjing1997@outlook.com (X.Z.); sunxc@shu.edu.cn (X.S.)

Abstract: Human head statistics is widely used in the construction of smart cities and has great market value. In order to solve the problem of missing pedestrian features and poor statistics results in a large-angle overlooking scene, in this paper we propose a human head statistics system that consists of head detection, head tracking and head counting, where the proposed You-Only-Look-Once-Head (YOLOv5-H) network, improved from YOLOv5, is taken as the head detection benchmark, the DeepSORT algorithm with the Fusion-Hash algorithm for feature extraction (DeepSORT-FH) is proposed to track heads, and heads are counted by the proposed cross-boundary counting algorithm based on scene segmentation. Specifically, Complete-Intersection-over-Union (CIoU) is taken as the loss function of YOLOv5-H to make the predicted boxes more in line with the real boxes. The results demonstrate that the recall rate and mAP@.5 of the proposed YOLOv5-H can reach up to 94.3% and 93.1%, respectively, on the SCUT_HEAD dataset. The statistics system has an extremely low error rate of 3.5% on the TownCentreXVID dataset while maintaining a frame rate of 18FPS, which can meet the needs of human head statistics in monitoring scenarios and has a good application prospect.

Keywords: YOLOv5-H; DeepSORT-FH; scene segmentation; cross-boundary counting; CIoU; overlooking scene

Citation: Wang, A.; Cao, X.; Lu, L.; Zhou, X.; Sun, X. Design of Efficient Human Head Statistics System in the Large-Angle Overlooking Scene. *Electronics* **2021**, *10*, 1851. https://doi.org/10.3390/electronics10151851

Academic Editor: Stefanos Kollias

Received: 16 June 2021
Accepted: 29 July 2021
Published: 31 July 2021

1. Introduction

With the advancement of science and technology, the construction of smart cities is the direction of future development. Intelligent video surveillance is an important part of a smart city, which needs to use Computer Vision (CV) algorithms to automatically analyze and process video content, and detect, track and identify objects in the videos in addition to capturing images. Human body detection is widely used in the security field of smart cities. However, security scenes are often complex; the density of the crowd and the complexity of the scenes lead to the fact that the human body object is likely to be occluded. Therefore, cameras in shopping malls, streets and other places are generally installed on the top of the environment, mostly from a bird's eye view, i.e., a large-angle overlooking scene. In this scene, the possibility of the human head being occluded is relatively small, and thus human body detection can be transformed into human head detection.

At present, there are mainly two schemes for the head detection. One is based on traditional image processing algorithms. Human heads can be detected by designing artificial features (e.g., gradient Histogram (Hog) [1], Harr descriptor [2]) and using classifiers that are commonly used by machine learning (e.g., Support Vector Machine (SVM) [3], Adaptive Boosting Tree (AdaBoost) [4]). However, the feature design of the traditional algorithms is very complicated, and the generalization ability of these algorithms is not strong, so it cannot adapt to multiple tasks. The other is based on the deep learning algorithms. Human heads can be detected by using Convolutional Neural Networks (CNNs) to learn object features automatically. The Region-CNN (R-CNN) proposed by Girshick [5] is a

pioneering work of an object detection algorithm based on deep learning. The R-CNN is based on the two-stage idea, where the first stage is to extract image features using CNNs, and the second stage is to extract candidate regions using the selective search method. The advantage of this algorithm is a high detection accuracy. However, the design of this algorithm is very complex and its processing speed is very slow. PENG et al. [6] proposed a DSCA network that uses a Feature Refinement Network (FRN) and a cascaded multi-scale architecture for feature extraction and regression, and has a better detection effect on small objects. However, the processing speed of this net-work is very slow. The authors of [7] proposed a model that is based on decoding an image for the detection of groups of people. They used a recurrent LSTM layer for sequence generation and trained the model end-to-end with a new loss function. This model has a good effect on object detection in crowded scenes. Redmon et al. [8] proposed a one-stage YOLO algorithm, which can directly return the location of the object to be detected from the input image, and the speed of detection can reach up to 45 fps, which greatly improves the object detection speed, but the detection accuracy is not ideal. Aiming at the deficiencies of the above algorithms, a series of improved algorithms have emerged one after another, such as SPP-Net [9] and Faster RCNN [10], which are based on the two-stage idea, and YOLOv2-YOLOv5 [11–13], SSD [14], etc., which are based on the one-stage idea.

Taking into account the problem of poor detection results due to the lack of pedestrian features in the large-angle overlooking scene, in this paper, we propose an efficient human head detection system that is composed of human head detection, multi-head tracking and head counting.

This paper makes the following major contributions:

1. We propose the YOLOv5-H network as the detection benchmark of the statistics system, where CIoU is taken as the loss function of the network to make the predicted boxes more in line with the real boxes;
2. We propose a fusion hash algorithm and include it in the DeepSORT [15] algorithm for feature extraction to track human heads;
3. The cross-boundary counting algorithm based on scene segmentation is proposed to count human heads;
4. We evaluate the detection performance of the improved YOLOv5-H on the SCUT_HEAD dataset and the statistics performance of the system on the TownCentreXVID dataset.

The structure of this paper is organized as follows. The related work and background are given in Section 2. Section 3 describes the proposed techniques for the human head statistics. Section 4 shows the experiments and results of our work. Finally, our work is concluded in Section 5.

2. Related Work and Background

2.1. Related Work

Many head detection approaches based on deep learning have been proposed. Vu et al. [16] proposed two context-aware CNN-based models. In this work, they focused on detecting human heads in natural scenes. Starting from the recent local R-CNN object detector, they extended it with two types of contextual cues, which can achieve state-of-the-art results, but they did not take into account motion information to perform long-term tracking. The authors of [17] introduced a novel end-to-end head detector called HeadNet, which is an adaptive relational framework with the local relation and global priors used to handle different changes of heads. Their experiments demonstrated that the HeadNet can achieve a high mAP@.5 of 91.3%. But in this work, the authors did not expand their method to consider the temporal–spatial relationship between video frames. In addition, multi-scale network training has not been explored. In [18], Gao et al. proposed a head detection approach based on a people-counting method combining the Adaboost algorithm and a CNN. Instead of adopting the general object region proposal method, they used the cascade Adaboost algorithm to obtain the head region proposals for the CNN, which can greatly reduce the operation time. However, they did not integrate tracking techniques into their method to handle the people-flow counting task.

In this paper, to address the above problems, the YOLOv5-H network, which adopts CIoU as the loss function to make the prediction boxes more in line with the real boxes and is then trained by multi scales, is proposed as the head detection benchmark. In addition, the DeepSORT-FH algorithm, which is improved from DeepSORT, is proposed to track the human heads. Finally, we propose a cross-boundary algorithm based on scene segmentation to handle the human head counting task more accurately.

2.2. YOLOv5

According to the depth of the network and the width of the feature map, YOLOv5 can be divided into four models, i.e., YOLOv5s, YOLOv5m, YOLOv5l and YOLOv5x, where YOLOv5s has the fastest processing speed and YOLOv5x has the highest detection accuracy. Specifically, the four models have the same network structure, which consists of input, backbone, neck and prediction. Figure 1 shows the structure of the YOLOv5s as a representative and each component of the network is described in the following subsections.

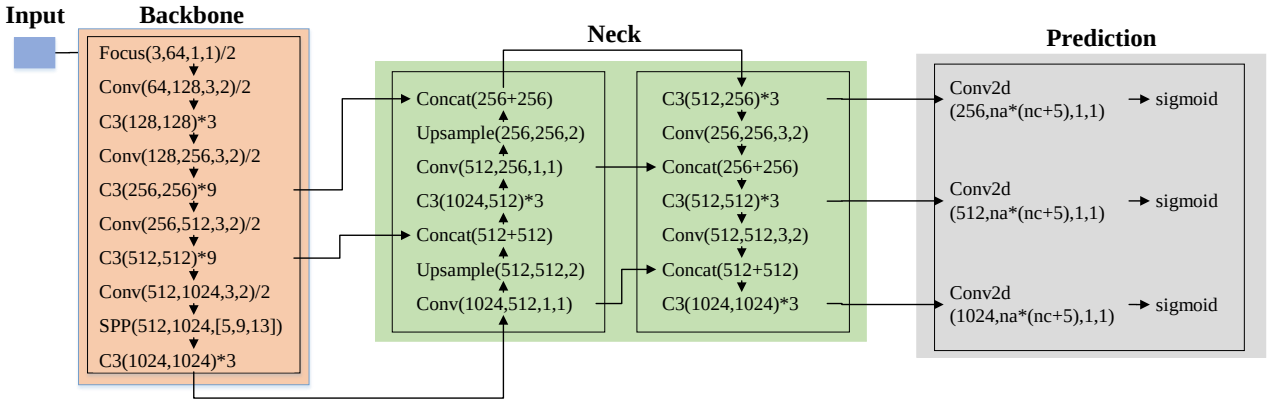

Figure 1. YOLOv5s network structure.

2.2.1. Input

During training, the Mosaic algorithm is used for data augmentation at the input end, which can improve the detection accuracy of small objects by randomly mixing four training images and stacking them with random size and distribution.

2.2.2. Backbone

Backbone is mainly composed of a focus structure and Cross-Stage-Partial-Network (CSPNet), where the the focus structure performs a slicing operation and CSPNet integrates the gradient change into the feature map, which can keep high accuracy while reducing the amount of calculation.

2.2.3. Neck

Neck uses a Feature Pyramid Network (FPN) and a Perceptual Adversarial Network (PAN). The FPN layer can convey the strong semantic features from the top to the bottom, and the PAN can convey the strong positioning features from the bottom to the top, so that the features can express strong semantic information.

2.2.4. Prediction

Prediction uses GIoU_Loss for the loss function of the bounding box.

2.3. DeepSORT

DeepSORT is a tracking framework based on a deep appearance feature model and a motion information model. After detecting the object, the trajectory of the object is predicted by the Kalman filter, which adopts a uniform and linear observation model. When the Kalman filter predicts the target's trajectory, the Mahalanobis distance between

the prediction box and the detection box is used as the matching metric of the two types of motion information.

However, using the Mahalanobis distance as the only scale for trajectory matching will cause uncertainty in the trajectory and increase the possibility of ID conversion. When the number of the tracked objects is large, a single Mahalanobis distance is often unable to effectively and accurately match the trajectory. Therefore, in addition to the motion information, for each tracked object will be extracted a deep appearance feature vector through the convolutional neural network, and a feature vector set is established for the same object. When the object moves for a long time, the appearance features of the same object may change, so by calculating the minimum cosine distance between the appearance feature of the new detection box and the appearance feature set of the matching object, the algorithm can judge whether the trajectory matches or not.

3. The Algorithm Architecture of Human Head Statistics System

Figure 2 shows the algorithm architecture of our proposed human head statistics system, which consists of YOLOv5-H for detection, DeepSORT-FH for tracking and a cross-boundary counting algorithm based on scene segmentation for statistics. Each part is described in the following subsections.

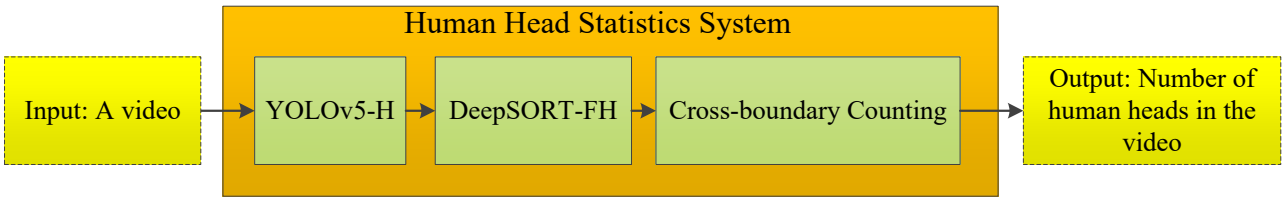

Figure 2. The algorithm architecture of human head statistics system.

3.1. Detection: YOLOv5-H

GIoU_Loss is used as the loss function of the original YOLOv5, as shown in Equation (1):

$$L_{GIoU} = 1 - (IoU - \frac{A_c - U}{A_c}) = 1 - (\frac{A \cap B}{A \cup B} - \frac{A_c - U}{A_c}), \quad (1)$$

where, firstly, IoU and the area of the smallest closed area of the two boxes are calculated, i.e., the area of the smallest box that contains both the prediction box and the real box. Secondly, the proportion of the closed area that does not belong to the two boxes in the closed area is calculated. Finally, IoU subtracts this proportion to obtain GIoU.

GIoU is not sensitive to scale and cannot solve the following situations:

- The prediction box is inside the real box;
- The prediction box is the same size as the real box.

In the large-angle overlooking scene, the detected heads are usually small, which results in that the above two situations happen frequently. However, CIoU can solve this problem, as shown in Equation (2):

$$L_{CIoU} = 1 - IoU + \frac{\rho^2(b, b^{gt})}{c^2} + \alpha\vartheta, \quad (2)$$

where α is the weight function, ϑ is used to measure the similarity of the aspect ratio, b and b^{gt} separately represent the center points of the prediction box and the real box, respectively, ρ is the Euclidean distance between the two center points, and c is the diagonal distance of the smallest closure area that can contain both the prediction box and the real box.

Therefore, CIoU_Loss is adopted as the loss function of YOLOv5-H in this paper, which can solve the problem that GIoU is not sensitive to scale by considering the distance to the center point of the bounding box and the aspect ratio of the bounding box.

3.2. Tracking: DeepSORT-FH

Multi-head tracking is a process of continuously marking the detected human heads. Figure 3 shows the flow of our proposed multi-head tracking algorithm, i.e., DeepSORT-FH. Firstly, the frame with bounding box output by YOLOv5-H is input into the motion information model and the appearance feature model. Secondly, the DeepSORT-FH algorithm uses the method of fusion measurement, that is, the similarity calculation and optimal incidence matrix solution are performed on the output of the two models. Finally, the head ID is the output of this algorithm. Specifically, in terms of appearance of the feature model, if deep features are extracted from each detected human head, the processing time of each frame will be greatly increased. To solve this problem, we propose a fusion hash algorithm to extract the features of human head quickly to replace the ReID feature extraction algorithm in the original DeepSORT.

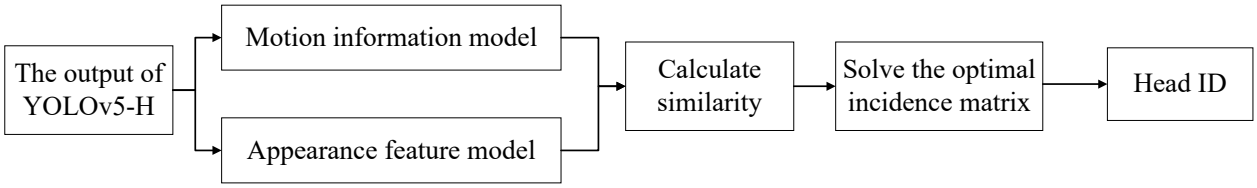

Figure 3. Multi-head tracking algorithm flow.

The fusion hash algorithm for feature extraction combines the average hash (aHash) algorithm and the perceptual hash (pHash) algorithm. Although the change range of the same head in adjacent frames is limited, the aHash algorithm is sensitive to optical flow, which can extract the color features and angle information of the human head and has better tracking performance for the same head. However, it is unstable to track different heads. The pHash algorithm is more robust, which can overcome the interference caused by a certain scale conversion, has better suppression of matching of different heads and can filter out different heads. However, its tracking of the same head is unstable, which may lead to the same head being missed. Therefore, we propose to combine the aHash with the pHash to make up for each other's shortcomings, which can greatly improve the accuracy of the head tracking.

Algorithm 1 shows the algorithm flow, i.e., the flow of solving the aHash and pHash feature vectors of one image. Firstly, the aHash algorithm transforms the human head into an 8×8 matrix by bicubic interpolation and the pHash uses Discrete Cosine Transform (DCT) to transform a human head image from the pixel domain to the frequency domain to obtain the low-frequency information of the image. The two-dimensional DCT is described in Equations (3) and (4):

$$F(u,v) = c(u)c(v)\sum_{i=0}^{N-1}\sum_{j=0}^{N-1} f(i,j)\, cos[\frac{(i+0.5)\pi}{N}u]\, cos[\frac{(i+0.5)\pi}{N}v] \tag{3}$$

$$c(u) = \begin{cases} \sqrt{\frac{1}{N}}, u = 0 \\ \sqrt{\frac{2}{N}}, u \neq 0 \end{cases} \tag{4}$$

where $f(i)$ is the original signal, $F(u)$ is the coefficient after DCT, N is the number of points of the original signal, and $c(u)$ is the compensation coefficient. The DCT matrix can be an orthogonal matrix. Secondly, the matrix output is greyed and flattened by aHash and only flattened by pHash. Thirdly, the weighted mean of the two matrices is calculated. Finally, the 64 matrix elements are compared with the weighted mean value. If the element is greater than the weighted mean value, it is recorded as 1, and otherwise it is recorded as 0. A vector of 64 elements is the output of the fusion hash Algorithm 1.

Algorithm 1 Fusion hash feature extraction algorithm.

Input: head image detected by Yolov5: M
Output: 128-dimensional feature vector: F
Process:

$r \leftarrow \text{imresize}(M, [8,8], 'bicubic')$
$g \leftarrow \text{rgb2gray}(r)$
$f_1 \leftarrow \text{flatten}(g)$
$d \leftarrow \text{dct}(g)$
$f_2 \leftarrow \text{flatten}(d)$
$mean\_value_{a/p} \leftarrow \text{mean}(f_{1/2})$
for $i \leftarrow 1$ *to* 64 **do**
  if $f_{1/2}(i) \geq mean\_value_{a/p}$ **then**
    $aHash(i) \leftarrow 1$
    $pHash(i) \leftarrow 1$
  else
    $aHash(i) \leftarrow 0$
    $pHash(i) \leftarrow 0$
  end if
end for
Return $F \leftarrow [aHash, pHash]$

In addition, in natural scenes, the object may leave the monitoring area and then return again, or the object may be occluded and lost during tracking. In response to the above situation, an object library is established to retain all the appearance features and motion information of the object in the first 70 frames, and introduce them into cascade matching, which effectively solves the problem that the ID does not switch after the object is temporarily lost, and improves the tracking accuracy.

3.3. Statistics: Cross-Boundary Counting Algorithm Based on Scene Segmentation

The general strategy of traffic statistics is to adopt a virtual line method that a virtual count line is set in the monitoring area and statistic analysis is performed by judging whether the object's motion trajectory crosses the line or not. Taking into account the randomness of the installation position of the surveillance camera and the randomness of the position where the object enters and leaves the monitoring area, it is difficult to accurately judge the various motions of the object by using the virtual line method. Therefore, as shown in Figure 3, we propose a cross-boundary counting algorithm based on a scene segmentation strategy. The monitoring area is divided into four sub-areas, i.e., A, B, C and D. Figure 4a depicts the general situation, as shown by the green arrows. When the object crosses the boundary between any two areas, that is, the sign of the difference between the horizontal or vertical coordinates of the center point of the adjacent position frames, one head is added to the number of heads.

Figure 4b describes the special situation that may happen. The green arrows indicates the situation in which the object crosses multiple sub-areas. For this situation, a de-duplication process is performed. When an object is counted, the algorithm will mask the path of this object in subsequent tracking, i.e., only counting each object once. The red arrows indicates that an object re-enters the monitoring area after leaving the monitoring area. If the object leaves the monitoring area for less than 70 frames, the de-duplication process will be performed, because the tracking algorithm can trace the object information of the previous 70 frames by cascading matching. Otherwise, the object will be regarded as a new object. The blue arrows indicate that the object does not pass the dividing line, i.e., entering and leaving in the same sub-area. For this situation, the algorithm will ignore it. The error rate caused by this situation can be reduced by increasing the number of sub-regions, but it will increase the processing time of the algorithm.

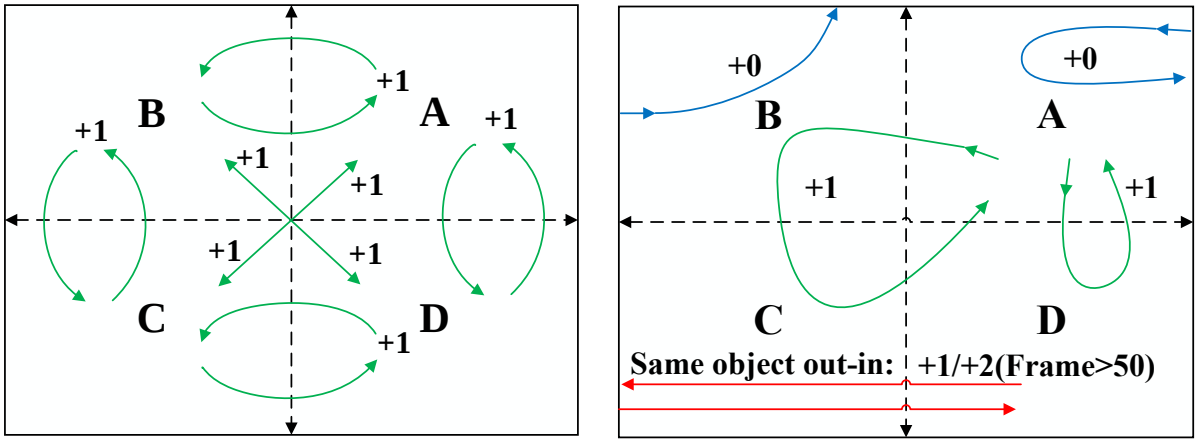

Figure 4. Cross boundary counting method based on scene segmentation: (**a**) describes the general situation; (**b**) describes the special situation.

4. Experimental Results and Discussion

The human head statistics system proposed in this paper was trained and tested on a platform based on Intel Xeon Gold 5218R CPU @ 2.1 GHz, 64 GB DDR4, NVIDIA Tesla T4 GPU with 16 GB video memory. The software environment was CUDA 10.1, Pytorch 1.5.0, Python 3.7.

4.1. Dataset

4.1.1. Image: SCUT_HEAD

SCUT_HEAD is a large-scale head detection dataset released by South China University of Technology in 2018. In this dataset, there are 4405 images in total and 111,251 head coordinates are annotated. The annotations of the dataset follow the PascalVOC standard. The dataset consists of two parts, i.e., part A and part B, where there are a total of 2000 images (training set: 1500 images, test set: 500 images) in part A, which are collected from classroom monitoring, and 67,321 human heads are labeled. The images in part B are all crawled from the Internet, containing a total of 2405 pictures (training set: 1905 images, test set: 500 images), in which 43,930 human heads are labeled.

4.1.2. Video: TownCentreXVID

The TownCentreXVID video is 5 min long and contains 25 frames sized 1920 × 1080 per second, i.e., a total of 7500 frames, of which the first 4500 frames are labeled and the last 3000 frames are not labeled.

4.2. Detection Accuracy of YOLOv5-H

In this study, YOLOv5s-H and YOLOv5m-H were trained on the SCUT_HEAD with 200 epochs at two scales of 640 × 640 and 1024 × 1024. As shown in Table 1, the accuracy of YOLOv5s-H@640 was stable at 0.880, the recall rate was stable at 0.904, mAP@.5 was stable at 0.891 and mAP@.5:.95 was stable at 0.450. The accuracy of YOLOv5m-H@640 was stable at 0.884, the recall rate was stable at 0.919, mAP@.5 was stable at 0.901, and mAP@.5:.95 was stable at 0.459. The accuracy of YOLOv5s-H@1024 was stable at 0.885, the recall rate was stable at 0.940, mAP@.5 was stable at 0.930, and mAP@.5:.95 was stable at 0.479. The accuracy of YOLOv5m-H@1024 was stable at 0.888, the recall rate was stable at 0.943, mAP@.5 was stable at 0.931, and mAP@.5:.95 was stable at 0.486, where mAP@.5 represents the mean average precision, and mAP@.5:.95 represents the mean average precision on different IoU thresholds (0.5, 0.55, 0.6, 0.65, 0.7, 0.75, 0.8, 0.85, 0.9, 0.95). The optimal weights of the two models were evaluated on a test set containing 22,300 human heads and 845 images, and the batch size was 8. The experimental results show that the operating speed of YOLOv5s-H in this test environment was about 2.3 times that of

YOLOv5m-H, but other accuracy indicators were worse than YOLOv5m-H. In addition, the results of large-scale (1024 × 1024) training were much higher than those of small-scale (640 × 640) training.

Table 1. Comparison with the test results of other algorithms in the SCUT_HEAD dataset.

Model	Accuracy Rate	Recall Rate	mAP@.5	mAP@.5:.95	Average Time
YOLOv5s-H@640	0.880	0.904	0.891	0.450	3.0 ms
YOLOv5m-H@640	0.884	0.919	0.901	0.459	5.7 ms
YOLOv5s-H@1024	0.885	0.940	0.930	0.479	7.0 ms
YOLOv5m-H@1024	0.888	0.943	0.931	0.486	15.0 ms

In order to further verify the performance of YOLOv5-H, it was compared with ReInspect, DSCA-Net, DSCA-Net + DSM, CFR-PHD and YOLOv5 on the SCUT_HEAD test set. As shown in Table 2, the YOLOv5-H outperformed other algorithms in both recall rate and mAP@.5, which are the key indicators. Specifically, the recall rate and mAP@.5 of YOLOv5 were slightly inferior to CFR-PHD and DSCA-Net+DSM, respectively. This is because heads are small in the large-angle overlooking scene and the YOLOv5 network adopts GIoU_Loss, which only contains the area information of the bounding box, which leads to the inaccurate regression of the network and insensitivity to small objects. However, the proposed YOLOv5-H adopts CIoU_Loss, which contains the overlapping area, the center point distance and the similarity of the aspect ratio between the real box and the prediction box, which leads to highly accurate regression of the network and improves sensitivity to small objects.

Table 2. Comparison with the test results of other algorithms in the SCUT_HEAD dataset.

Model	Accuracy Rate	Recall Rate	mAP@.5
ReInspect [7]	0.80	0.86	0.78
DSCA-Net [6]	0.88	0.86	0.87
DSCA-Net + DSM	0.91	0.88	0.89
CFR-PHD [19]	0.89	0.91	0.877
YOLOv5s@1024	0.87	0.882	0.878
YOLOv5m@1024	0.884	0.902	0.886
YOLOv5s-H@1024	0.885	0.94	0.930
YOLOv5m-H@1024	0.888	0.943	0.931

4.3. The Accuracy of Human Head Statistics System

The proposed human head statistics system was tested on the TownCentreXVID video dataset. In this paper, we only count the human heads, so the evaluation indicators are the actual number of heads and the number of heads counted by the system. As shown in Table 3, the statistics system that only relies on detection and tracking had a large error due to missing or wrong detection and object occlusion, which cause the frequent switching IDs of the tracking algorithm. However, after including the cross-boundary counting algorithm based on scene segmentation, the number of falsely counted heads was greatly reduced. This is because in a statistics systems without the cross-boundary algorithm, whether the number of heads increases by 1 depends on ID switching, whereas in a statistics system with the cross-boundary algorithm, whether the number of heads increases by 1 depends on whether the head crosses the boundary, which means that only when ID switching and cross boundary occur at the same time, the accuracy of the system will be negatively affected, but this situation rarely happens. In conclusion, the statistics system with the cross-boundary algorithm can effectively suppress the problems of leakage, false detection and ID switching, which greatly improves the robustness of the system.

In addition, the results show that the error rate of the statistics system using YOLOv5m-H was 15.2% lower than that of using YOLOV5s, but the frame rate was only reduced by

3 fps, and thus our proposed system finally adopts the implementation of YOLOv5m-H + DeepSORT-FH + cross-boundary counting.

Table 3. The performance of this paper's head statistics algorithm on the TownCentreXVID dataset.

Method	Actual Heads	Counted Heads	Error Counted Heads	Error Rate	Frames
YOLOv5s-H + DeepSORT-FH	230	521	291 more	165.2%	21FPS
YOLOv5s-H + DeepSORT-FH + Cross-boundary	230	273	43 more	18.7%	21FPS
YOLOv5m-H + DeepSORT-FH	230	455	225 more	97.8%	18FPS
YOLOv5m-H + DeepSORT-FH + Cross-boundary	230	238	8 more	3.5%	18FPS

5. Conclusions

In this paper, we proposed a human head statistics system that is composed of YOLOv5-H as the detection benchmark, DeepSORT-FH for head tracking, and a cross-boundary counting algorithm based on scene segmentation for head statistics, where YOLOv5-H is trained at multiples scale and uses CIoU_Loss as the loss function to make the predicted boxes more in line with the real boxes. The DeepSORT-FH adopts the fusion hash algorithm as an appearance feature model to extract head features quickly. On the SCUT_HEAD dataset, the recall rate can reach up to 94.3% and the mAP@.5 can reach up to 93.1%. Our proposed system had an error of 3.5% on the TownCentreXVID video dataset while maintaining a frame rate of 18 fps, which can meet the needs of head statistics in monitoring scenarios and has a good application prospect.

Author Contributions: Conceptualization, A.W. and X.C.; methodology, L.L.; software, X.Z.; validation, L.L., X.S. and X.Z.; formal analysis, A.W.; investigation, A.W.; resources, X.C.; data curation, X.Z.; writing—original draft preparation, X.Z.; writing—review and editing, A.W.; visualization, X.S.; supervision, L.L.; project administration, L.L.; funding acquisition, A.W. All authors have read and agreed to the published version of the manuscript.

Funding: This research was funded by the National Key Research and Development Program of China, grant number 2017YFB0403502.

Acknowledgments: The authors would like to thank the reviewers and editors for their hard work on this research.

Conflicts of Interest: The authors declare no conflict of interest.

References

1. Dalal, N.; Triggs, B. Histograms of oriented gradients for human detection. In Proceedings of the 2005 IEEE Computer Society Conference on Computer Vision and Pattern Recognition (CVPR'05), San Diego, CA, USA, 20–25 June 2005; Volume 1, pp. 886–893.
2. Viola, P.; Jones, M. Rapid object detection using a boosted cascade of simple features. In Proceedings of the 2001 IEEE Computer Society Conference on Computer Vision and Pattern Recognition, CVPR 2001, Kauai, HI, USA, 8–14 December 2001; Volume 1, p. I.
3. Chang, C.C.; Lin, C.J. LIBSVM: A library for support vector machines. *Acm Trans. Intell. Syst. Technol. (TIST)* **2011**, *2*, 1–27. [CrossRef]
4. Li, X.; Wang, L.; Sung, E. AdaBoost with SVM-based component classifiers. *Eng. Appl. Artif. Intell.* **2008**, *21*, 785–795. [CrossRef]
5. Girshick, R.; Donahue, J.; Darrell, T.; Malik, J. Rich Feature Hierarchies for Accurate Object Detection and Semantic Segmentation. In Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition (CVPR), Washington, DC, USA, 23–28 June 2014.
6. Peng, D.; Sun, Z.; Chen, Z.; Cai, Z.; Xie, L.; Jin, L. Detecting Heads using Feature Refine Net and Cascaded Multi-scale Architecture. In Proceedings of the 2018 24th International Conference on Pattern Recognition (ICPR), Beijing, China, 20–24 August 2018; pp. 2528–2533. [CrossRef]
7. Stewart, R.; Andriluka, M.; Ng, A.Y. End-To-End People Detection in Crowded Scenes. In Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition (CVPR), Las Vegas, NV, USA, 27–30 June 2016.
8. Redmon, J.; Divvala, S.; Girshick, R.; Farhadi, A. You Only Look Once: Unified, Real-Time Object Detection. In Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition (CVPR), Las Vegas, NV, USA, 27–30 June 2016.

9. He, K.; Zhang, X.; Ren, S.; Sun, J. Spatial Pyramid Pooling in Deep Convolutional Networks for Visual Recognition. *IEEE Trans. Pattern Anal. Mach. Intell.* **2015**, *37*, 1904–1916. [CrossRef] [PubMed]
10. Ren, S.; He, K.; Girshick, R.; Sun, J. Faster R-CNN: Towards Real-Time Object Detection with Region Proposal Networks. *arXiv* **2016**, arXiv:cs.CV/1506.01497.
11. Redmon, J.; Farhadi, A. YOLO9000: Better, Faster, Stronger. *arXiv* **2016**, arXiv:cs.CV/1612.08242.
12. Redmon, J.; Farhadi, A. YOLOv3: An Incremental Improvement. *arXiv* **2018**, arXiv:cs.CV/1804.02767.
13. Bochkovskiy, A.; Wang, C.Y.; Liao, H.Y.M. YOLOv4: Optimal Speed and Accuracy of Object Detection. *arXiv* **2020**, arXiv:cs.CV/2004.10934.
14. Wei, L.; Dragomir, A.; Dumitru, E.; Christian, S.; Scott, R.; Cheng-Yang, F.; Alexander, B. SSD: Single Shot MultiBox Detector. In *Computer Vision—ECCV 2016*; Springer: Amsterdam, The Netherlands, 2016.
15. Veeramani, B.; Raymond, J.W.; Chanda, P. DeepSort: Deep convolutional networks for sorting haploid maize seeds. *BMC Bioinform.* **2018**, *19*, 289. [CrossRef] [PubMed]
16. Vu, T.H.; Osokin, A.; Laptev, I. Context-Aware CNNs for Person Head Detection. In Proceedings of the IEEE International Conference on Computer Vision (ICCV), Santiago, Chile, 7–13 December 2015.
17. Li, W.; Li, H.; Wu, Q.; Meng, F.; Xu, L.; Ngan, K.N. HeadNet: An End-to-End Adaptive Relational Network for Head Detection. *IEEE Trans. Circuits Syst. Video Technol.* **2020**, *30*, 482–494. [CrossRef]
18. Gao, C.; Li, P.; Zhang, Y.; Liu, J.; Wang, L. People counting based on head detection combining Adaboost and CNN in crowded surveillance environment. *Neurocomputing* **2016**, *208*, 108–116. CrossRef]
19. Jie, Z.; Li, C.; Zheng, L.; Sen, W.; Ze, C. Pedestrian head detection algorithm based on clustering and Faster RCNN. *J. Northwest Univ. (Natural Sci. Ed.)* **2020**, *50*, 971–978. (In Chinese)