Faster, Lighter, Robuster: A Weakly-Supervised Crowd Analysis Enhancement Network and A Generic Feature Extraction Framework

Shaokai Wu[1], Zhaogeng Liu[2], Wencheng Pei[1], Jianbo Hong[1], Zhanshan Li[1]
[1]Collage of Computer Science and Technology, Jilin University
[2]School of Artificial Intelligence, Jilin University
{wusk2419, zgliu20, peiwc9919, hongjb2419}@mails.jlu.edu.cn, lizs@jlu.edu.cn

Abstract

With bounding box labels needed for training, object detection is viewed unfavorably in terms of crowd analysis, due to the intensive labor for labeling and the unsatisfactory performance in clutters and severe occlusions. Another feasible method, density-based regression, despite its proficiency in counting and only point-level labels used for training, cannot get the location of each person, and the time and space consumption is relatively high. In this paper, we propose a generic feature extraction framework, Adaptive Pyramid Score (APS), based on object detection and designed specifically for extracting quantitative and spatial-semantic features. Moreover, as an intuitive and feasible solution regarding crowd analysis, we propose the weakly-supervised Confidence-Threshold-Foresight Network (CTFNet) under our APS feature extraction framework, which only needs count-level labels for training and improves the performance of various methods dramatically. Our system realizes the triple enhancement of counting, localization, and detection, which is also proved to be faster than advanced crowd analysis methods, lighter to be transplanted to various object detection methods, and robuster to tackle tasks of extreme scenes. Furthermore, the weakly-supervised paradigm leverage the intensive labor for labeling profoundly.

1. Introduction

Crowd analysis mainly includes crowd counting, crowd localization, and face detection, which has always been a heated yet challenging task in computer vision, for its close relationship with humans and the difficulty of detecting under severe occlusions and clutters [12, 21, 33]. However, most cutting-edge methods focus on a single task such as either counting or localization and have a high time and space consumption together with a strong dependence on fully labeled data. The lack of new systems makes it hard to push forward further analysis such as crowd tracking, anomalous activity detection, etc [15, 27].

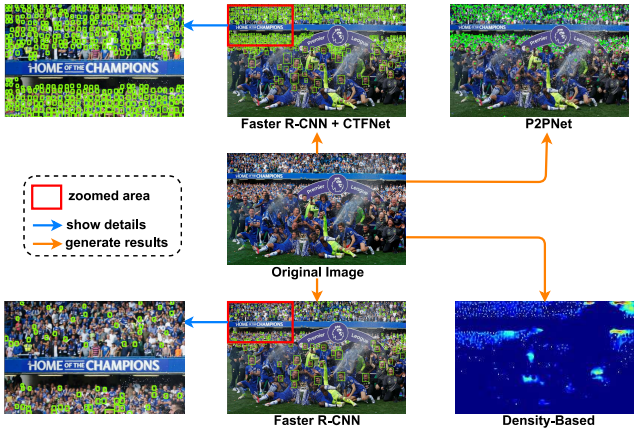

Figure 1. Illustrations for the comparison of our method with existing methods, in which the crowds are marked in Green. **Bottom right:** The mainstream density-based methods fail to offer the precise locations of individuals. **Top right:** P2PNet cannot provide the detection box of individuals. **Directly below:** The traditional detection-based methods are unable to obtain satisfactory detection results in dense crowd scenes, with a zoom-in on their details at the bottom left. **Directly above:** Our method, which predicts the accurate counts while providing precise locations and scale information, is zoomed in on its details in the upper left corner.

Although object detection can give precise locations and detection boxes of individuals, it cannot tackle the counting tasks of crowded scenes and is strongly dependent on bounding box labels, as viewed unfavorably by most relevant papers such as [2,8,12,19,21,27,33]. Another feasible solution, which is point-based, P2PNet [27], can obtain the point locations of human heads even in extremely occluded scenes but fails to meet the need of providing head detection boxes and has a challenging time and space consumption.

On the other hand, current advanced crowd analysis methods are mostly density-based [21], which perform relatively satisfactorily when tackling counting tasks. They work by summing over the predicted density maps to get

the count of crowds and offer a heat map for visualization. However, these methods have difficulty in offering precise locations of individuals, and it is hard for them to obtain superior ground truth density maps by adopting a Gaussian kernel [16, 21] to the point annotations.

We regard locating individuals with boxes as a more fine-grained solution than only point locations or a single count, as getting detection boxes naturally solves the triple tasks of counting, localization, and detection. Further, the boxes can be used for other downstream tasks such as crowd tracking and face recognition. Despite the urgent need of crowd analysis in this increasingly-crowded world, especially in the global pandemic of Covid-19, there seem to be no methods that can gather the merits of all crowd analysis methods while bypassing their drawbacks. That makes us wonder *how good it would be if there existed a detection system that is faster, lighter, and robuster than all previous methods*! Motivated by this, we determine to design a novel detection system to tackle crowd analysis.

After extensive research and experiments, we propose a weakly-supervised enhancement system for object detection in crowd analysis, aimed at solving two crucial problems of object detection: the feature loss in high-density crowds and the single detection means in extremely varied scenes. To tackle these problems, we propose the Adaptive-Pyramid-Score (APS) feature extraction framework to extract deeper features from object detectors and resolve the feature loss problem. Then we propose the Confidence-Threshold-Foresight Network (CTFNet) to make full use of APS features, which needs only count-level labels for training and greatly enhances the performance of object detection in crowd analysis. The successful transplantation on YOLO-Series [6, 24], Faster R-CNN [25], and LSC-CNN [26] manifests the compatibility of our system. The visualization of various methods is shown in Figure 1.

We highlight our main contributions as follows:

1. We propose a generic feature extraction framework APS, which can extract multiple features and have strong compatibility with various methods.

2. We propose a **weakly-supervised** crowd analysis system CTFNet that can enhance various detection methods. To the best of our knowledge, this is the first generic enhancement system in crowd analysis.

3. Our system has been proved to be **faster** than classic crowd analysis methods, **lighter** to be transplanted to various methods, and **robuster** to tackle the analysis of all crowd density with **counting**, **localization**, and **detection**.

2. Related Works

2.1. Density-Based Methods

Since they were first proposed in [16], the density-based crowd analysis methods with CNN have been continuously ameliorated. They have gradually become the mainstream while achieving outstanding results [21, 32, 33]. These methods [19, 20, 23, 28, 31] obtain the counts by summing over the estimated density maps. To bypass labor-intensive point-level annotations, some weakly-supervised methods based on density maps are also proposed. However, they fail in offering precise locations [27] and the exact sizes of individuals. Besides, their time and space consumption is highly dependent on the size of the input image [30].

2.2. Object Detection Methods

As one of the earliest ideas to tackle crowd analysis, object detection provides the count of the crowd while giving fine-grained estimation, i.e., locations of individuals along with exact scales of heads. Besides, object detection performs a dramatic detection accuracy in sparse scenarios [5, 17, 22, 26, 39]. On the contrary, the result will go unsatisfactory under the conditions of high occlusions and clutters. With bounding box [27] labels needed for training, there remains a drudgery regarding time and labor for labeling.

2.3. Feature Extraction Framework

Lately, some newly proposed feature extractors have extensively drawn the public's attention. With proposals of VIT [4], DETR [1], and other methods, the transformer, which was first introduced into Natural Language Processing (NLP) [3, 29], is gradually applied in computer vision. In the meanwhile, the multi-scale feature-extraction method Feature Pyramid Networks (FPN) [18] benefits the detection of smaller objects and is continuously improved [14, 38, 40]. This all shows the great potential of the new feature extraction approach.

3. Our Work

Overview: The brief architecture of our weakly-supervised enhancement system is shown in Figure 5, consisting of APS feature extraction framework and CTFNet.

First, multiple features are extracted by an object detector with APS Framework (Section 3.1). Then these features are flattened and concatenated, forming the input feature of CTFNet (Section 3.2). The internal of CTFNet is the Confidence Layer, Threshold Layer, and Foresight Layer, all of which consist of fully connected layers.

The function of C, T, F Layers is to adaptively change the detection threshold and detection means (Section 3.3). With this property, object detection methods perform better in both sparse and dense crowds. The learning targets are all generated by count-level labels, with the weak supervision manner detailedly discussed in Section 3.2.

Notation: $[\ \ ]$ represents the Iverson bracket. $\boldsymbol{\mathit{Italic\ bold}}$ indicates the matrix. $\lfloor\rfloor$ means the floor function.

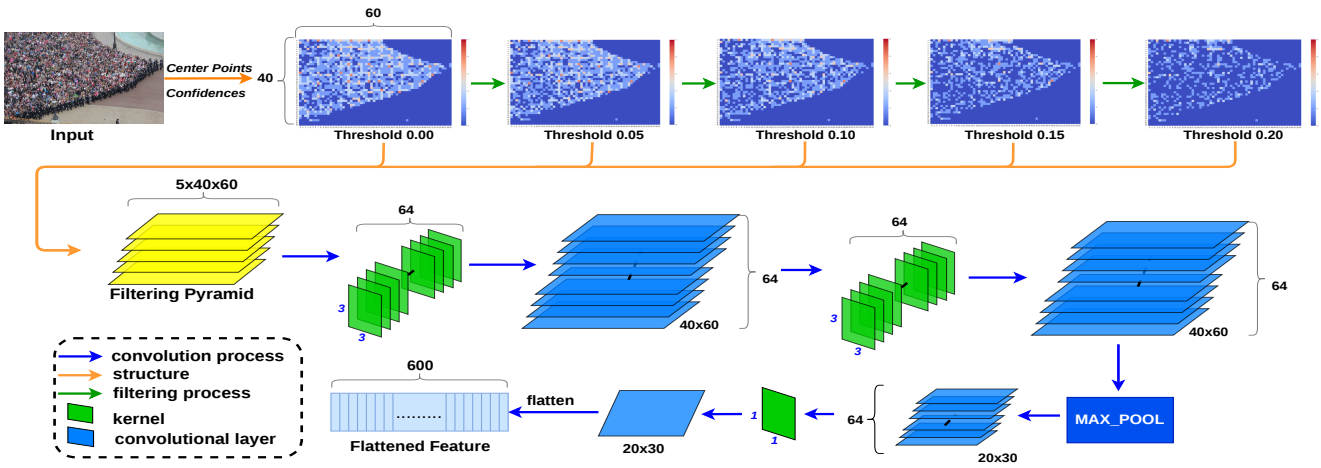

Figure 2. Illustration for the feature extraction process of Filtering Pyramid. The top row demonstrates the filtering and mapping effect of the five-layer Filtering Pyramid. Considering that the multi-class Filtering Pyramid is a four-dimensional matrix, we decide to take the crowd, which is single-class, as an example for visualization. Each heat map represents a filtering layer, demonstrating locations and counts of detection boxes at different thresholds. The middle and bottom rows demonstrate the convolution, pooling, and flattening process of the Filtering Pyramid.

3.1. APS Feature Extraction Framework

APS feature extraction framework is to extract deeper features from object detectors, avoiding conflict with the internal implementation of specific methods. This unique quality enables APS Framework to be easily transplanted to various methods.

The Filtering Pyramid

Whether it is Faster R-CNN [25], YOLO [24], or other object detection methods, eventually the foreground objects are detected with detection boxes and confidence scores. Commonly, a threshold is used to filter the objects of low confidence. However, the fixed threshold screening method ignores the difference between images. Specifically, the object with a low confidence score in a high foreground density scene may be the foreground to be detected, while a non-foreground area with few foregrounds still has the chance to be given a high confidence score. Thus, the use of a single threshold to identify the foregrounds within a picture is prone to low-density false detection and high-density missed detection. Besides, this approach ignores the spatial-semantic information, i.e., the relative location of each center point of the object and the foreground object distribution in the picture. To tackle this problem, we propose the Filtering Pyramid to conduct a deeper extraction of quantitative and spatial-semantic features.

First, we map the center points of detected foreground objects on a rectangle grid, by their relative locations to the original image and its confidence score. The top row of Figure 2 demonstrates this mapping process. To be specific, assume the count of floors in the Filtering Pyramid is 5, then all the objects' center points will be mapped on a grid, according to 5 different thresholds. Notice that the higher the threshold setting, the sparser the mapped grid, because objects with low confidence scores are filtered.

The ***Pyramid*** refers to the Filtering Pyramid, containing the number of foregrounds of each class in each location and at every threshold. The number of boxes is n, the width and height of the input image are w_0 and h_0, and the number of wide and high grids are w and h. With the box index as x, its score and class are $s_{[x]}$ and $c_{[x]}$, the location of its horizontal and vertical coordinates are $w_{[x]}$ and $h_{[x]}$. We denote $Pyramid_{[t][k][i][j]}$ as the number of foregrounds in the k-th class under the threshold of Filtering Pyramid indexed by t, with the width and height of the grid indexed by i and j. The formula is shown below.

$$Pyramid_{[t][k][i][j]} = \sum_{x=0}^{n-1} [s_{[x]} \geq t \wedge c_{[x]} = k \wedge \lfloor w_{[x]} / \frac{w_0}{w} \rfloor = i \wedge \lfloor h_{[x]} / \frac{h_0}{h} \rfloor = j] \quad (1)$$

After that, we use convolutional layers to further process the Filtering Pyramid, with ReLU activation used after each convolutional layer. Take the five-layer Filtering Pyramid as an example, assuming that its grid width, height, and the number of foreground classes are h, w, and c, so its shape becomes $5 \times c \times h \times w$. We use 64 convolutional kernels to increase the Filtering Pyramid to 64 channels, and then performed a 64 to 64 channels convolution, followed by max-pooling. Eventually, the pyramid is scaled to 1 channel by a convolutional kernel with 64 channels, as shown in the middle and bottom rows of Figure 2.

BoxScore

To further extract quantitative features, the confidence scores of all objects are taken into account. Through ex-

tensive research, we discover that images with dense foregrounds tend to possess more detection boxes that have low confidence scores and vice versa. Therefore, the scores of foregrounds reflect the quantitative features of the whole image to a certain extent. To improve the generalization of extracted features, the scores should be distributed in the interval [0, 1]. For a multi-class condition, the softmax function is used to make the scores of each class sum up to 1. Note that the softmax function is not done to a single class and makes the scores in this class averaged, instead it is done over all classes to normalize the confidence score of each class. This process can be simplified into the sigmoid process when there is only one class, on which condition the sigmoid function is used to normalize the distribution of scores in this single class into the interval [0, 1].

The ***BoxScore*** is a matrix that comprises foreground distribution of different confidence scores. The number of the detection boxes and the number of classes are n and c, respectively. For the i-th $(0 \leq i < n)$ box, the score of its j-th $(0 \leq j < c)$ class is $s_{[i][j]}$. The number of score groups is g. The scores are distributed between 0 and 1 after softmax processing, so the score group width is $\frac{1}{g}$. We denote $BoxScore_{[k][j]}$ as the number of ***BoxScore*** in the k-th$(0 \leq k < g)$ score interval and the j-th class.

$$BoxScore_{[k][j]} = \sum_{i=0}^{n-1} [\frac{k}{g} \leq s_{[i][j]} < \frac{k+1}{g}] \qquad (2)$$

RPNAnchor

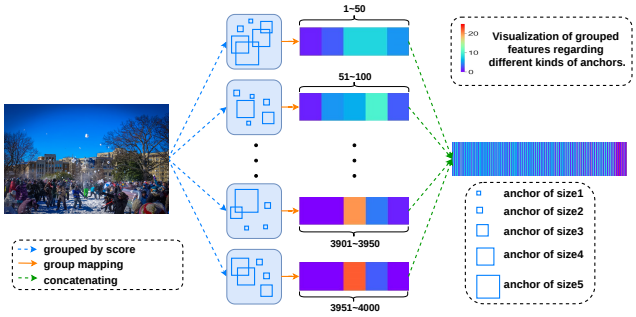

Figure 3. Distribution of different scales of candidate boxes at different confidences.

Region Proposal Network (RPN) [25] is widely used in some object detection methods. Yet we find that more features can be extracted from RPN to make full use of it.

The distribution of different candidate boxes is a reflection of both quantitative and spatial features since they have different ratios and scales [25]. Specifically, the image with dense foreground generally has small object sizes, and its predicted candidate boxes focus on small scales. In contrast, a picture with sparse foreground renders the candidate boxes to concentrate on large scales. To find more representative features, we fixedly select the top N = 4000 highest scoring candidate boxes and compute their distribution by groups. Suppose we have 5 scales of candidate boxes and we use 80 confidence groups. We count the number of each of these 5 scales of candidate boxes among the 50 anchors in each group. The ***RPNAnchor*** matrix is formed by concatenating these groups of numbers, as shown in Figure 3.

Let s be the number of anchor scales. The number of confidence groups is g, and the group interval is $\frac{N}{g}$. The anchor serial number is k $(0 \leq k < N)$, and $a_{[k]}$ represents the anchor scale. We denote $RPNAnchor_{[i][j]}$ as the i-th $(0 \leq i < g)$ group and j-th $(0 \leq j < s)$ scale anchor of ***RPNAnchor***.

$$RPNAnchor_{[i][j]} = \sum_{k=n/g\times i}^{n/g\times(i+1)-1} [a_{[k]} = j] \qquad (3)$$

RPNScore

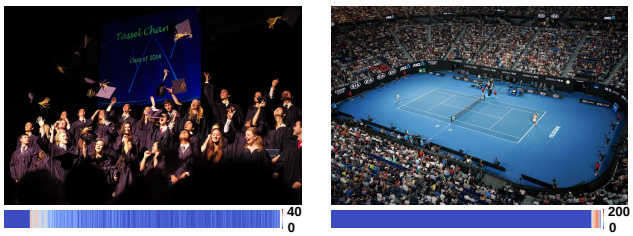

Figure 4. The spectrum under each image is the distribution of foreground density at different confidence scores.

Different from the ***BoxScore*** mentioned before, the confidence scores given by RPN only distinguish between two classes, i.e., foreground and background, so the foreground score is the one we need careful analysis.

We still select the top N = 4000 proposals with the highest scores, and the distribution of their scores is essentially a reflection of the quantitative features. In sparse scenarios, the number of foreground objects is likely to be less than N, and many objects with low scores (background objects) are selected, so the scores are scattered. Otherwise, the scores are centered at a high level as the foreground objects are abundant. The visualization is in Figure 4.

For a better generalization, the scores of candidate boxes also need to be normalized into the interval [0, 1], so the sigmoid function is used for normalization again. The number of candidate boxes is counted according to their scores, forming the score distribution of candidate boxes.

Let N be the number of selected candidate boxes, and the number of groups dividing scores is g. The scores are distributed in the interval [0, 1] after normalization, so the group width is $\frac{1}{g}$. The index of the current proposal is i $(0 \leq i < N)$ and its score is $s_{[i]}$, then the score feature of the j-th $(0 \leq j < g)$ group in ***RPNScore*** is:

$$RPNScore_{[j]} = \sum_{i=0}^{N-1} [\frac{j}{g} \leq s_{[i]} < \frac{j+1}{g}] \qquad (4)$$

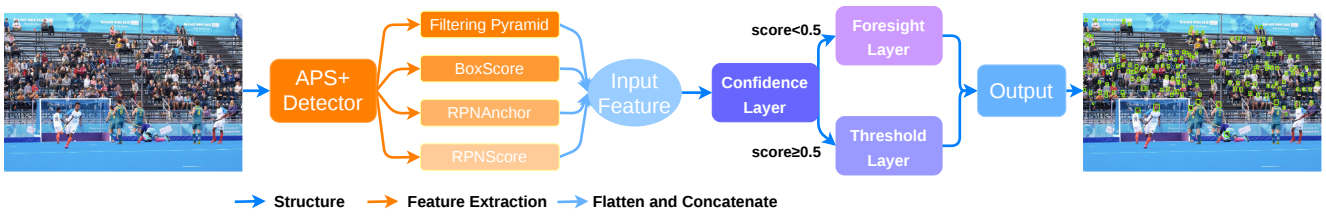

Figure 5. Illustration for the CTFNet structure. The input image first enters an object detector with APS Framework, from which features are extracted. Then features are flattened and concatenated to form a final input feature, which enters the Confidence Layer to obtain the confidence score and decides to enter the Threshold Layer or Foresight Layer according to the score.

3.2. Weakly-Supervised Training CTFNet

Confidence-Threshold-Foresight Network is an enhancement crowd analysis network for object detection methods, which needs only count-level labels for training. CTFNet consists of Confidence Layer (C Layer), Threshold Layer (T Layer), and Foresight Layer (F Layer). APS Framework and CTFNet make up the whole architecture of our enhancement system, as shown in Figure 5.

In the next three subsections, we illustrate how CTFNet can train with only count-level labels, and why it is the weakly-supervised paradigm. Notice that this section focuses on the weakly-supervised training paradigm. And the functions of C, T, F Layers are shown in Section 3.3.

Input Feature

The input feature is formed by concatenating all flattened APS features. According to a specific method, the total of extracted features can be different. To be specific, all object detection methods can extract at least two APS features, i.e., ***FilteringPyramid*** and ***BoxScore***. While methods with RPN can additionally extract ***RPNAnchor*** and ***RPNScore***. All extracted features are firstly flattened to one-dimensional and then concatenated to form the input feature of CTFNet.

Automatically Labeled Output Value

The output values of C, T, F Layers are a confidence score, a threshold value, and a count of crowds, respectively. The label of count has been given by count-level datasets, and the confidence score and threshold value are labeled automatically as shown in the next two paragraphs.

Confidence score labels are generated by measuring whether a detector can get the ground truth count by adjusting the detection threshold. Notice that the relation between the threshold and detection count is a monotone function, and the maximum and minimum count can be fetched by setting the threshold to 0 and 1. Thus, if the ground truth count is not between the maximum and minimum count, it means the object detector can not get the ground truth count by detection, and the confidence label will be marked as 0. Otherwise, it will be marked as 1.

Threshold value labels are marked by binary searches to find an optimum threshold. Specially, if the ground truth count is higher than the maximum detection count, the threshold label will be marked as 0, i.e., not filtering detected boxes. If lower than the minimum detection count, it will be marked as 1 to block all boxes. Except for these two special cases, on other occasions the threshold label is marked by binary search, i.e., threshold values 0 and 1 are used as two endpoints, and the binary search will stop when and only when the detection count is equal (or very nearly) to ground-truth count.

Weakly-Supervised Paradigm

The training paradigm of CTFNet is more weakly-supervised than self-supervised. Although the use of APS Features makes it seem like self-supervised learning, it should be noticed that CTFNet is an enhancement framework independent of object detection methods, and APS features originate from object detection. Furthermore, the training need of CTFNet is count-level labels, and others are labeled automatically. Therefore, CTFNet is a weakly-supervised enhancement network.

3.3. The Function of CTF

In this part, the functions of C, T, F Layers are shown. The internal of each layer is fully connected layers, as described in Hyperparameter settings. The input feature and output value follow that in Section 3.2.

The Confidence Layer

The function of C Layer is to make choice between T Layer and F Layer. Usually, the scene is not extremely crowded and using T Layer to adjust the detection threshold works well. On very rare occasions that classic object detection can not tackle properly, such as the fourth scene in Figure 8, F Layer will be chosen to handle it.

The output value is a float number of the score. T Layer is chosen if the score is above 0.5, else F Layer is chosen. Binary cross-entropy loss is adopted as the loss function.

The Threshold Layer and ThreshLoss

The T Layer is proficient in adaptively adjusting threshold values. We discover that the threshold of many popular object detection methods such as [1, 6, 25] are fixed. However, images with dense foregrounds require a lower threshold to improve the recall rate while images with sparse foregrounds need a higher threshold to bypass false positives. With the help of the T Layer, CTFNet can effectively deal

with most scenes, such as the first three in Figure 8.

To improve its performance, we design the ThreshLoss as the loss function for the T Layer. After thorough experiments, we notice that the smaller the threshold, the greater the impact on the recall rate. For example, increasing the threshold from 0 to 0.1 brings a much larger fluctuation in the number of detection boxes than raising it from 0.5 to 0.6. Thus, the impact brought by the initial threshold should be taken into consideration when designing the loss function. We decide to adopt the form of the square of the error divided by the true threshold to represent the loss function. Also, to prevent numerical explosion due to a too-small denominator, we add an offset α to the denominator. y is the predicted value and y' is the threshold label value.

$$ThreshLoss(y, y') = (y - y')^2/(y' + \alpha) \qquad (5)$$

The Foresight Layer

The F Layer is used to foresee the count of crowds by regression. The motivation comes from that many previous researches [2, 19, 27] pointed out that object detection methods fail to cope with occlusions and clutters. Although T Layer has solved most of the problems they mentioned, object detection still performs unsatisfactorily in extreme scenes. While our F Layer can compensate for this deficiency.

When using the F Layer, the counting result will be the estimation result and Mean Squared Error is used as the loss function during training. For a better detection performance, the detection threshold will be turned to 0 to fit the large count of crowds. And then the binary searches should be done to optimize the Non-Maximum suppression (NMS) threshold. To be specific, we set the upper limit of the NMS threshold β = 0.3. The binary search uses the current NMS threshold and β as two endpoints and will cease when the number of detection boxes is equal (or very close) to the estimation result of the F Layer. Specially, if the estimation result is greater than the number of boxes at NMS threshold β, the binary search will not start, on which condition the final NMS threshold will be set as β.

4. Experiments

Hyperparameters: In the APS framework, the detailed parameters of the Filtering Pyramid follow that in Figure 2. The number is 400 for both BoxScore groups and RPN-Score groups. For RPNAnchor, the number of different anchor scales is 5 and the number of its groups is 80. In CTFNet, the offset α in ThreshLoss is 0.05. C, T, F Layers have the same structure, consisting of 3 fully connected layers interleaved with ReLU activations. Besides the input and output layer, there is one hidden layer with 256 hidden nodes. Adam is used as the optimizer with a learning rate of 3e-5.

Evaluation Metrics: Following the mainstream methods, we adopt Mean Absolute Error (MAE) and Root Mean Squared Error (RMSE) as counting evaluation metrics, GAME [7] metric as localization metrics, and F1 Measure, Precision, and Recall as detection metrics.

Datasets: Extensive experiments are conducted on five publicly challenging datasets, including ShanghaiTech PartA (SH_A) and PartB (SH_B) [37], UCF_CC_50 [10], UCF_QNRF [11], NWPU-Crowd [34] and Wider Face [36].

Computing Power: Limited by resources, our training device is a laptop consisting of R7-4800H and Nvidia 2060. Most experiments are done on it except the time and space experiments in Section 4.5, in which a server with Nvidia 3090 is temporarily used.

4.1. Enhancement performance for counting

Experiment settings: All methods are trained on the same dataset NWPU-Crowd [34] for fair comparisons.

Since our enhancement system can be easily transplanted to various methods, we show the enhancement degree by a method without and with our system, of which the latter is prefixed with 'CTF'. The compatibility of our system with FPN [18] is also tested. The counting enhancement on SH_A and UCF_QNRF datasets is shown in Figure 6.

Notice that the performance of classic methods is strongly dependent on the setting of the threshold, so we measure their performance under multiple thresholds. With the enhancement of our system, the methods are no longer subject to the threshold, so the results manifest as points, with horizontal lines added for better visualizations.

1. After adding our system to classic methods, the MAE is lessened ranging from 34.3% (YOLOX [6] in SH_A) to 53.4% (Faster R-CNN in UCF_QNRF), compared with even the best-fine-tuned threshold version.

2. Our APS feature extraction framework can work with other feature extraction frameworks such as FPN. And our system proves to work better compared with a single FPN when applied to Faster R-CNN. These all show the strong compatibility of our enhancement system.

3. Object detection methods are no longer subjected to threshold settings in terms of crowd analysis, which shows more possibility of tackling severe situations adaptively.

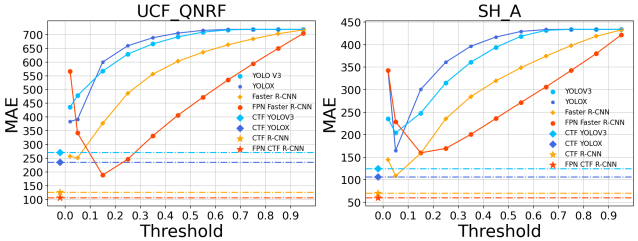

Figure 6. Illustrations of the enhancement of various object detection methods on two counting datasets UCF_QNRF and SH_A with CTFNet applied.

Method	Venue	*Result*			*UCF_CC_50*		*SH_A*		*SH_B*		*UCF_QNRF*	
		Count	Location	Size	MAE ↓	RMSE ↓	MAE ↓	RMSE ↓	MAE ↓	RMSE ↓	MAE ↓	RMSE ↓
LSC-CNN [26]	PAMI21	✓	✓	✓	225.6	302.7	66.4	117.0	8.1	12.7	120.5	218.2
Bayesian+ [21]	ICCV19	✓	✗	✗	229.3	308.2	62.8	101.8	7.7	12.7	88.7	154.8
S-DCNet [35]	ICCV19	✓	✗	✗	204.2	301.3	58.3	95.0	6.7	10.7	104.4	176.1
DM-Count [33]	NerIPS20	✓	✗	✗	211.0	291.5	59.7	95.7	7.4	11.8	85.6	**148.3**
AMSNet [9]	ECCV20	✓	✗	✗	208.4	297.3	56.7	93.4	6.7	10.2	101.8	163.2
ASNet [13]	CVPR20	✓	✗	✗	174.8	251.6	57.8	90.1	-	-	91.6	159.7
SUA-Fully [23]	ICCV21	✓	✗	✗	-	-	66.9	125.6	12.3	17.9	119.2	213.3
P2PNet [27]	ICCV21	✓	✓	✗	172.7	256.2	**52.7**	85.1	**6.3**	9.9	**85.3**	154.5
CTF-LSC (Ours)	-	✓	✓	✓	**168.3**	**224.6**	53.4	**82.3**	7.1	**9.7**	90.8	166.7

Table 1. Comparisons in counting, in which the current SOTAs and runner-ups are marked with **bold** and underline, respectively.

4.2. Comparisons with SOTAs in counting

Experiment settings: All compared methods are trained on corresponding training datasets for fair comparisons, and follow the 5-fold cross-validation on UCF_CC_50 [10].

As a rare object detection method in crowd analysis, LSC-CNN [26] can get the location and size of individuals beyond a single count, which many mainstream approaches cannot. But its counting performance loses to that of advanced density-based and point-based methods. However, our enhancement system makes it great again! After adding our system to LSC-CNN, it successfully outstrips current state-of-the-art methods in several benchmarks. The MAE is lowered by 20.5% and RMSE is lowered by 26.2% on average, as shown in Table 1.

4.3. Enhancement performance for localization

Experiment settings: The same as that of Section 4.1.

Method	*SHA*			*UCF_QNRF*		
	GAME(1) ↓	GAME(2) ↓	GAME(3) ↓	GAME(1) ↓	GAME(2) ↓	GAME(3) ↓
YOLOV3	216.8	233.1	259.8	454.3	471.0	505.3
YOLOX	170.9	187.2	223.5	405.6	421.7	462.9
Faster R-CNN	116.6	131.0	157.5	264.1	279.6	308.5
CTF YOLOV3	139.1	173.3	215.8	287.9	323.5	396.3
CTF YOLOX	113.2	139.6	184.6	252.1	278.3	334.2
CTF R-CNN	68.4	90.5	127.4	119.6	143.8	201.0

Table 2. Enhancement of CTFNet on Localization

Our system can also improve the localization performance of various methods. GAME [7] is an existing metric used as evaluation of localization performance, which is defined as $GAME(L) = \frac{1}{N} \cdot \sum_{i=1}^{N} (\sum_{l=1}^{4^L} |e_i^l - gt_i^l|)$, where N is total of images, e_i^l and gt_i^l represent estimated count and ground truth count of l-th region in i-th image, and L is the restrict factor. The results are shown in Table 2.

4.4. Enhancement performance for detection

Experiment settings: Models are trained on the Wider Face training dataset and evaluated on the validation dataset by F1 Measure, Precision, and Recall, where IOU is 0.5.

Benefiting from adaptively changing threshold and detection means, the performance of Faster R-CNN in F1 Measure and Precision is also greatly enhanced with little influence on the Recall rate, as shown in Table 3.

Set	*Faster R−CNN*			*CTF R−CNN*		
	F1 Measure ↑	Precision ↑	Recall ↑	F1 Measure ↑	Precision ↑	Recall ↑
Easy	11.14%	5.92%	93.59%	20.03%	11.24%	91.98%
Medium	18.43%	10.24%	91.97%	29.29%	17.46%	90.88%
Hard	32.24%	19.95%	83.98%	46.16%	32.09%	82.17%

Table 3. Enhancement of CTFNet on Detection. The validation dataset is officially split into 3 parts, easy, medium, and hard.

4.5. Time Consumption and Space Consumption

Experiment settings: All compared methods follow their settings in the papers or code, and the evaluation dataset is UCF_QNRF.

The consumption of time and space is closely related to whether it can work on low-performance devices. We evaluated them by the mean processing time and mean used video memory per image, as shown in Figure 7.

Compared with other crowd analysis methods, object detection has a relatively low time and space consumption. Although classic object detection performs unsatisfactorily in dense crowds, this deficiency has been compensated in an all-round way by our brand new system.

Benefiting from the light structure of APS Framework and CTFNet, there is almost no additional consumption after adding our system, especially in terms of space.

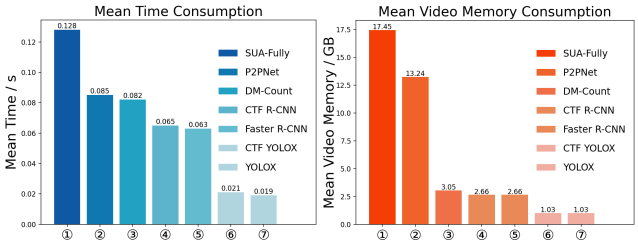

Figure 7. Time Consumption and Space Consumption. Methods are ranked by consumption from high to low in each chart.

4.6. Ablation Studies

Each part of our weakly-supervised system is evaluated to prove its effectiveness. Here we show 13 group ablation studies of the counting performance on the SH_A dataset. The experiment setting is identical to Section 4.1.

APS features: Faster R-CNN is selected as the detector to show the enhancement degree of each APS Feature

Figure 8. The robustness in tackling analysis tasks towards all densities of crowds, including counting, localization, and detection. The detection results are generated by CTF R-CNN, with the count of crowds in the bottom-right corner of each image.

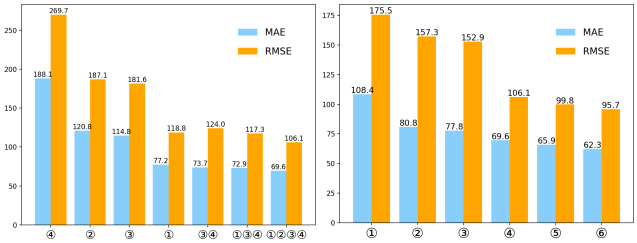

Figure 9. Illustrations of the effect of each component in our weakly-supervised system.

i.e., ***FilteringPyramid***, ***BoxScore***, ***RPNAnchor*** and ***RPNScore*** on SH_A dataset, and only the F Layer is used to follow the univariate analysis. Here we assign numbers ①, ②, ③, and ④ for the above four features in turn.

With the number of features rising, MAE and RMSE are decreasing. The best enhancement is achieved when all 4 features are available, as shown in the left part of Figure 9.

CTFNet: Except ① is Faster R-CNN without our system, the rest are enhanced by CTFNet. ② uses only the T Layer with MSE loss while ③ uses the T Layer with ThreshLoss. ④ uses only the F Layer. ⑤ and ⑥ both use all CTF Layers, with the former using MSE loss in the T Layer and the latter using ThreshLoss.

T Layer (② and ③) brings down the error of Faster R-CNN. ④ displays the enhancement with only F Layer applied. ⑤ and ⑥ exhibit the effect of using all three layers of CTF with the error further reduced. Further, the improvement from ②, ⑤ to ③, ⑥ manifests the effect of ThreshLoss. The results are shown in the right part of Figure 9.

4.7. Faster, Lighter, Robuster

Multi-Dimensional enhancements of our system have been proved with extensive experiments. It is still worthwhile to be noticed that our system needs only count-level labels for training while achieving triple enhancement in counting, localization, and detection.

As for 'Faster', Figure 7 has shown the processing time advantages. Besides, it takes a very short time for training even on our laptop with Nvidia 2060. This merit enables us to do extensive experiments with limited computing resources.

'Lighter' is the capability to be easily transplanted to various methods, which we regard as the essential quality of our enhancement system. The successful transplantation on various methods manifests the lightness of our weakly-supervised system. Further, the little additional space consumption in Figure 7 also shows this valuable quality.

Finally, 'Robuster' has double meanings. The first is the capability to enhance all parts of crowd analysis, including counting (Section 4.1), localization (Section 4.3), and detection (Section 4.4). The second is the robustness in detecting all densities of crowds, as shown in Figure 8.

5. Conclusion

In this work, we propose a weakly-supervised enhancement system to improve the performance of various methods in crowd analysis with only count-level labels. The motivation comes from the phenomenon that advanced methods concentrate on either counting or localization while neglecting the overall performance in crowd analysis. Our system includes APS Feature Extraction Framework and CTFNet, all of which are easy to be transplanted with strong compatibility. After adding our enhancement system, crowd analysis becomes more all-rounded, achieving a multi-dimensional improvement in counting, localization, and detection. Compared with state-of-the-art methods, the enhanced methods under our system have a comprehensive performance with a low time and space consumption. Faster, Lighter, Robuster, all of which are realized with only count-level labels, which leverage the intensive labor for labeling and enhance the performance of various methods profoundly.

References

[1] Nicolas Carion, Francisco Massa, Gabriel Synnaeve, Nicolas Usunier, Alexander Kirillov, and Sergey Zagoruyko. End-to-end object detection with transformers. In Andrea Vedaldi, Horst Bischof, Thomas Brox, and Jan-Michael Frahm, editors, *Computer Vision - ECCV 2020 - 16th European Conference, Glasgow, UK, August 23-28, 2020, Proceedings, Part I*, volume 12346, pages 213–229, 2020. 2, 5

[2] Zhi-Qi Cheng, Jun-Xiu Li, Qi Dai, Xiao Wu, and Alexander G. Hauptmann. Learning spatial awareness to improve crowd counting. In *2019 IEEE/CVF International Conference on Computer Vision, ICCV 2019, Seoul, Korea (South), October 27 - November 2, 2019*, pages 6151–6160, 2019. 1, 6

[3] Zihang Dai, Zhilin Yang, Yiming Yang, Jaime G. Carbonell, Quoc Viet Le, and Ruslan Salakhutdinov. Transformer-xl: Attentive language models beyond a fixed-length context. In *Proceedings of the 57th Conference of the Association for Computational Linguistics, ACL 2019, Florence, Italy, July 28- August 2, 2019, Volume 1: Long Papers*, pages 2978–2988, 2019. 2

[4] Alexey Dosovitskiy, Lucas Beyer, Alexander Kolesnikov, Dirk Weissenborn, Xiaohua Zhai, Thomas Unterthiner, Mostafa Dehghani, Matthias Minderer, Georg Heigold, Sylvain Gelly, Jakob Uszkoreit, and Neil Houlsby. An image is worth 16x16 words: Transformers for image recognition at scale. In *9th International Conference on Learning Representations, ICLR 2021, Virtual Event, Austria, May 3-7, 2021*, 2021. 2

[5] Weina Ge and Robert T. Collins. Marked point processes for crowd counting. In *2009 IEEE Computer Society Conference on Computer Vision and Pattern Recognition (CVPR 2009), 20-25 June 2009, Miami, Florida, USA*, pages 2913–2920, 2009. 2

[6] Zheng Ge, Songtao Liu, Feng Wang, Zeming Li, and Jian Sun. YOLOX: exceeding YOLO series in 2021. *CoRR*, abs/2107.08430, 2021. 2, 5, 6

[7] Ricardo Guerrero-Gómez-Olmedo, Beatriz Torre-Jiménez, Roberto Javier López-Sastre, Saturnino Maldonado-Bascón, and Daniel Oñoro-Rubio. Extremely overlapping vehicle counting. 9117:423–431, 2015. 6, 7

[8] Yutao Hu, Xiaolong Jiang, Xuhui Liu, Baochang Zhang, Jungong Han, Xianbin Cao, and David S. Doermann. Nascount: Counting-by-density with neural architecture search. In *Computer Vision - ECCV 2020 - 16th European Conference, Glasgow, UK, August 23-28, 2020, Proceedings, Part XXII*, volume 12367, pages 747–766, 2020. 1

[9] Yutao Hu, Xiaolong Jiang, Xuhui Liu, Baochang Zhang, Jungong Han, Xianbin Cao, and David S. Doermann. Nascount: Counting-by-density with neural architecture search. 12367:747–766, 2020. 7

[10] Haroon Idrees, Imran Saleemi, Cody Seibert, and Mubarak Shah. Multi-source multi-scale counting in extremely dense crowd images. In *2013 IEEE Conference on Computer Vision and Pattern Recognition, Portland, OR, USA, June 23-28, 2013*, pages 2547–2554, 2013. 6, 7

[11] Haroon Idrees, Muhmmad Tayyab, Kishan Athrey, Dong Zhang, Somaya Al-Máadeed, Nasir M. Rajpoot, and Mubarak Shah. Composition loss for counting, density map estimation and localization in dense crowds. In *Computer Vision - ECCV 2018 - 15th European Conference, Munich, Germany, September 8-14, 2018, Proceedings, Part II*, volume 11206, pages 544–559, 2018. 6

[12] Xiaoheng Jiang, Li Zhang, Mingliang Xu, Tianzhu Zhang, Pei Lv, Bing Zhou, Xin Yang, and Yanwei Pang. Attention scaling for crowd counting. In *2020 IEEE/CVF Conference on Computer Vision and Pattern Recognition, CVPR 2020, Seattle, WA, USA, June 13-19, 2020*, pages 4705–4714, 2020. 1

[13] Xiaoheng Jiang, Li Zhang, Mingliang Xu, Tianzhu Zhang, Pei Lv, Bing Zhou, Xin Yang, and Yanwei Pang. Attention scaling for crowd counting. pages 4705–4714, 2020. 7

[14] Alexander Kirillov, Ross B. Girshick, Kaiming He, and Piotr Dollár. Panoptic feature pyramid networks. In *IEEE Conference on Computer Vision and Pattern Recognition, CVPR 2019, Long Beach, CA, USA, June 16-20, 2019*, pages 6399–6408, 2019. 2

[15] Yinjie Lei, Yan Liu, Pingping Zhang, and Lingqiao Liu. Towards using count-level weak supervision for crowd counting. *Pattern Recognit.*, 109:107616, 2021. 1

[16] Victor S. Lempitsky and Andrew Zisserman. Learning to count objects in images. In *Advances in Neural Information Processing Systems 23: 24th Annual Conference on Neural Information Processing Systems 2010. Proceedings of a meeting held 6-9 December 2010, Vancouver, British Columbia, Canada*, pages 1324–1332. Curran Associates, Inc., 2010. 2

[17] Sheng-Fuu Lin, Jaw-Yeh Chen, and Hung-Xin Chao. Estimation of number of people in crowded scenes using perspective transformation. *IEEE Trans. Syst. Man Cybern. Part A*, 31(6):645–654, 2001. 2

[18] Tsung-Yi Lin, Piotr Dollár, Ross B. Girshick, Kaiming He, Bharath Hariharan, and Serge J. Belongie. Feature pyramid networks for object detection. In *2017 IEEE Conference on Computer Vision and Pattern Recognition, CVPR 2017, Honolulu, HI, USA, July 21-26, 2017*, pages 936–944, 2017. 2, 6

[19] Weizhe Liu, Mathieu Salzmann, and Pascal Fua. Context-aware crowd counting. In *IEEE Conference on Computer Vision and Pattern Recognition, CVPR 2019, Long Beach, CA, USA, June 16-20, 2019*, pages 5099–5108, 2019. 1, 2, 6

[20] Xiyang Liu, Jie Yang, Wenrui Ding, Tieqiang Wang, Zhijin Wang, and Junjun Xiong. Adaptive mixture regression network with local counting map for crowd counting. In Andrea Vedaldi, Horst Bischof, Thomas Brox, and Jan-Michael Frahm, editors, *Computer Vision - ECCV 2020 - 16th European Conference, Glasgow, UK, August 23-28, 2020, Proceedings, Part XXIV*, volume 12369, pages 241–257, 2020. 2

[21] Zhiheng Ma, Xing Wei, Xiaopeng Hong, and Yihong Gong. Bayesian loss for crowd count estimation with point supervision. In *2019 IEEE/CVF International Conference on Computer Vision, ICCV 2019, Seoul, Korea (South), October 27 - November 2, 2019*, pages 6141–6150, 2019. 1, 2, 7

[22] Zheng Ma, Lei Yu, and Antoni B. Chan. Small instance detection by integer programming on object density maps. In *IEEE Conference on Computer Vision and Pattern Recognition, CVPR 2015, Boston, MA, USA, June 7-12, 2015*, pages 3689–3697, 2015. 2

[23] Yanda Meng, Hongrun Zhang, Yitian Zhao, Xiaoyun Yang, Xuesheng Qian, Xiaowei Huang, and Yalin Zheng. Spatial uncertainty-aware semi-supervised crowd counting. In *2021 IEEE/CVF International Conference on Computer Vision, ICCV 2021, Montreal, QC, Canada, October 10-17, 2021*, pages 15529–15539. IEEE, 2021. 2, 7

[24] Joseph Redmon, Santosh Kumar Divvala, Ross B. Girshick, and Ali Farhadi. You only look once: Unified, real-time object detection. In *2016 IEEE Conference on Computer Vision and Pattern Recognition, CVPR 2016, Las Vegas, NV, USA, June 27-30, 2016*, pages 779–788, 2016. 2, 3

[25] Shaoqing Ren, Kaiming He, Ross B. Girshick, and Jian Sun. Faster R-CNN: towards real-time object detection with region proposal networks. *IEEE Trans. Pattern Anal. Mach. Intell.*, 39(6):1137–1149, 2017. 2, 3, 4, 5

[26] Deepak Babu Sam, Skand Vishwanath Peri, Mukuntha Narayanan Sundararaman, Amogh Kamath, and R. Venkatesh Babu. Locate, size, and count: Accurately resolving people in dense crowds via detection. *IEEE Trans. Pattern Anal. Mach. Intell.*, 43:2739–2751, 2021. 2, 7

[27] Qingyu Song, Changan Wang, Zhengkai Jiang, Yabiao Wang, Ying Tai, Chengjie Wang, Jilin Li, Feiyue Huang, and Yang Wu. Rethinking counting and localization in crowds: A purely point-based framework. In *Proceedings of the IEEE/CVF International Conference on Computer Vision*, pages 3365–3374, 2021. 1, 2, 6, 7

[28] Guolei Sun, Yun Liu, Thomas Probst, Danda Pani Paudel, Nikola Popovic, and Luc Van Gool. Boosting crowd counting with transformers. *CoRR*, abs/2105.10926, 2021. 2

[29] Ashish Vaswani, Noam Shazeer, Niki Parmar, Jakob Uszkoreit, Llion Jones, Aidan N. Gomez, Lukasz Kaiser, and Illia Polosukhin. Attention is all you need. In *Advances in Neural Information Processing Systems 30: Annual Conference on Neural Information Processing Systems 2017, December 4-9, 2017, Long Beach, CA, USA*, pages 5998–6008, 2017. 2

[30] Aditya Vora. FCHD: A fast and accurate head detector. *CoRR*, abs/1809.08766, 2018. 2

[31] Jia Wan and Antoni B. Chan. Modeling noisy annotations for crowd counting. In Hugo Larochelle, Marc'Aurelio Ranzato, Raia Hadsell, Maria-Florina Balcan, and Hsuan-Tien Lin, editors, *Advances in Neural Information Processing Systems 33: Annual Conference on Neural Information Processing Systems 2020, NeurIPS 2020, December 6-12, 2020, virtual*, 2020. 2

[32] Jia Wan, Ziquan Liu, and Antoni B. Chan. A generalized loss function for crowd counting and localization. In *IEEE Conference on Computer Vision and Pattern Recognition, CVPR 2021, virtual, June 19-25, 2021*, pages 1974–1983, 2021. 2

[33] Boyu Wang, Huidong Liu, Dimitris Samaras, and Minh Hoai Nguyen. Distribution matching for crowd counting. In *Advances in Neural Information Processing Systems 33: Annual Conference on Neural Information Processing Systems 2020, NeurIPS 2020, December 6-12, 2020, virtual*, 2020. 1, 2, 7

[34] Qi Wang, Junyu Gao, Wei Lin, and Xuelong Li. Nwpu-crowd: A large-scale benchmark for crowd counting and localization. *IEEE Trans. Pattern Anal. Mach. Intell.*, 43(6):2141–2149, 2021. 6

[35] Haipeng Xiong, Hao Lu, Chengxin Liu, Liang Liu, Zhiguo Cao, and Chunhua Shen. From open set to closed set: Counting objects by spatial divide-and-conquer. pages 8361–8370, 2019. 7

[36] Shuo Yang, Ping Luo, Chen Change Loy, and Xiaoou Tang. WIDER FACE: A face detection benchmark. pages 5525–5533, 2016. 6

[37] Yingying Zhang, Desen Zhou, Siqin Chen, Shenghua Gao, and Yi Ma. Single-image crowd counting via multi-column convolutional neural network. In *2016 IEEE Conference on Computer Vision and Pattern Recognition, CVPR 2016, Las Vegas, NV, USA, June 27-30, 2016*, pages 589–597, 2016. 6

[38] Gangming Zhao, Weifeng Ge, and Yizhou Yu. Graphfpn: Graph feature pyramid network for object detection. In *Proceedings of the IEEE/CVF International Conference on Computer Vision (ICCV)*, pages 2763–2772, 2021. 2

[39] Tao Zhao and Ramakant Nevatia. Bayesian human segmentation in crowded situations. In *2003 IEEE Computer Society Conference on Computer Vision and Pattern Recognition (CVPR 2003), 16-22 June 2003, Madison, WI, USA*, pages 459–466, 2003. 2

[40] Xiaodong Zhao, Junliang Chen, Minmin Liu, Kai Ye, and Linlin Shen. Multi-scale attention-based feature pyramid networks for object detection. In *Image and Graphics - 11th International Conference, ICIG 2021, Haikou, China, August 6-8, 2021, Proceedings, Part I*, volume 12888 of *Lecture Notes in Computer Science*, pages 405–417, 2021. 2