CIS-680: Final Project Report: Counting Machine Parts

Ajay Anand, Ankit Billa, Benedict Florance Arockiaraj, Elizabeth Dinella

Abstract

*Counting objects in an image is a task applicable across many domains. For instance, crowd counting, inventory counting, and cell counting have been the focus of recent research. The major challenges in estimating the count of objects include overlapping objects, object scale issues, occlusions, and varying lighting conditions. In this report, we explore the problem of counting machine washer parts. Our technique is an extension of FamNet [10] with an additional loss component, trained on the given dataset. We compare to three baseline methods: a traditional image processing pipeline, instance segmentation, and density map estimation. We evaluate the performance of these algorithms by computing the Mean Absolute Error (MAE) and the Root Mean Squared Error (RMSE) between the true object counts and the model outputs. Our approach achieves a performance of **1.96 MAE**.*

1. Introduction

The task of object counting has a wide range of applications such as crowd estimation, inventory management, and other industrial processes. Each domain suffers from its own unique challenges, but densely packed objects pose a problem to many. In this report, we study the task of counting densely packed objects.

Given that object detection systems have made great strides in recent years, they may seem to be a natural solution to object counting. However, our experiments in Section **??** find the current state of the art object detection systems to struggle with counting densely packed objects.

To resolve these issues, state of the art techniques have been developed for the task of counting objects in dense environments, focusing on crowd estimation with the goal of increasing crowd safety. In our work, we utilize these algorithms for counting machine parts given their baseline performance in low-light environments with many overlapping objects.

Our approach builds on FamNet [10], a regression based approach which learns a density map for each image. FamNet uses a Mean Squared Error loss function between the true and predicted density maps. However, this loss function does not take into the account the localized density map error, and only considers errors in the total. In order to solve this problem, we propose an alternative loss function called *mismatch* loss. Since our dataset includes 9 angles for each set of objects, we take the average of predicted counts for each angle, as a post processing step.

We evaluate our approach against several traditional and deep-learning based baselines. We also include experiments on out of distribution data sets to evaluate our techniques generalizability.

2. Related Work

2.1. Traditional Approaches

Traditional methods for machine vision operations such as product counting, error control and dimension measurement mainly revolve around image filtering techniques to obtain object counts or even detect objects. One such pipeline was proposed by Baygin *et al*. [2], which combines multiple image filtering techniques with thresholding methods to obtain edge maps of each circular object, and then counts the number of circles generated using a Hough Transform.

Since the given dataset specifically consists of circular machine parts, we implement the same pipeline to get baseline results.

2.2. Object Detection

There has been much work in object detection [12, 5], in which the goal is to identify and locate objects in an image. Object detection approaches employ a sliding window to localize objects within a small local region. Initial naive approaches to object counting simply run an object detection model and count the number of detected objects. This naive approach is somewhat effective in low density scenes, but has trouble generalizing to images with a high density of objects. Furthermore, architectures that employ sliding windows are heavyweight and computationally expensive.

Advances in using whole image detection techniques remedy some of the drawbacks of sliding window based techniques [12, 15] but still have the same drawbacks as object detection algorithms for detecting objects in dense or obscured images.

Regression based approaches [8, 11, 15] address the poor performance of object detection systems to count ob-

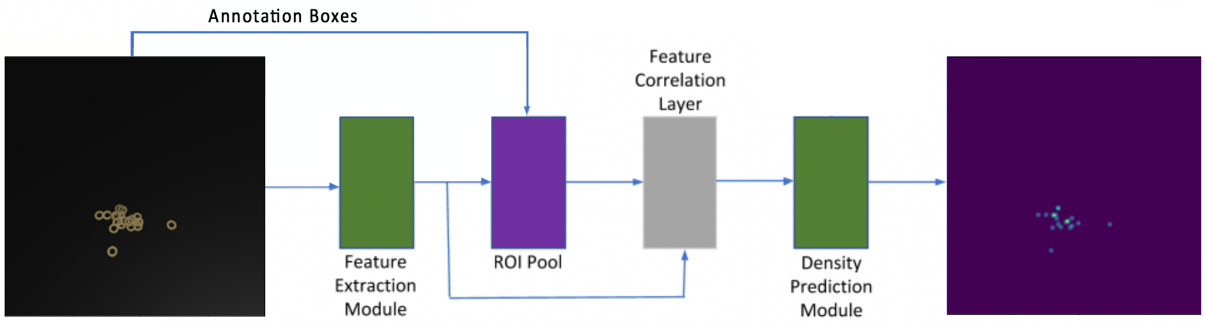

Figure 1. Proposed Model Architecture based on FamNet [10]

jects in densely packed scenes. These methods attempt to learn the count directly without locating and counting each object. They directly learn a map between image features and a number of objects. These image features may be local or whole image.

2.3. Density Estimation

Recent contributions in the field of object counting have focused on the task of density estimation [6, 1]. These approaches incorporate spatial information and also side-step the difficult task of learning to detect and localize each individual object. However, these techniques are more well suited for large crowds, where the exact count is not needed. For our setting of machine part counting, the number of objects is much less and thus the margin of error should be lower.

We see a shift in the overall algorithmic approach from traditional to CNN-based methods for object counting. Specifically, recent works use multi-scale features to tackle the scale variation problem [15], attention based methods to focus on useful information [5] and use auxiliary tasks [13] to improve counting performance.

3. Dataset Analysis

The dataset consists of 8100 images of washer parts in a bundle, with 900 unique images and each image being captured from 9 different angles. The images are of dimensions 1080×1080, with the annotations file having bounding boxes and masks for every image in COCO format. Processing the `annotation.json` file in the dataset shows that the average object count per image is 25.969 and the median object count per image is 21.

4. Proposed Method

We base our proposed method off of [10], which uses a few-shot regression based model to perform object counting. They use the image, along with a few exemplar object boxes from the image, to predict a density map, that estimates the count of objects in the image.

4.1. Ground Truth Density Construction

To generate the density map, we first use Gaussian smoothing with a dynamic window size dependent on the dataset. Deviating from [10], where the dataset is specifically annotated point-wise, we instead fix the point annotations to the center-of-mass of the object polygon. Having computed the point annotations, we calculate the average distance between every point and its 1-nearest neighbor. The computed average distance acts as the dynamic window size for the Gaussian smoothing. We use the same hyperparameters for smoothing (std dev $\sigma = window\_size/4$) as mentioned in [10].

4.2. Network Architecture

We use the FamNet architecture proposed in [10], with slight adjustments to the overall structure. Figure 1 shows the pipeline of the network. The network consists of two components: a multi-scale feature extraction module and a density prediction module. The feature extraction component uses pre-trained ImageNet ResNet-50 [4] backbone for feature extraction. We make a number of modifications to FamNet for machine part counting.

FamNet, is unique in that it can adapt to counting any particular class at test time, regardless of the class label. Given only a few exemplar bounding boxes, the model can adapt to counting objects of that class. As its title, *Learning to Count Everything* suggests, FamNet is highly generalizable to different domains.

In our setting, we are not concerned with adapting to different classes at test time. At training and inference time, we have a single class (washer) which we aim to count. Rather than give exemplar bounding boxes of washers, as in the original paper, we give all bounding boxes around each washer. In order to accommodate for this, we obtain multi-scale features of the bounding boxes by performing ROI pooling on the convolutional feature maps. The convolutional feature maps after the fourth block of the backbone is passed to the density estimation module.

To make the density prediction module category agnostic, the authors take a correlation between the pooled feature map from the objects and the image feature extraction map. The density estimation module has a series of five convolution blocks and three upsampling layers interleaved. The 2D density map is obtained using a final 1×1 convolution layer. By summing the density map, we can obtains the floating-point count of the number of objects in the input image.

4.3. Angle Aggregation

The provided washer dataset includes multiple perspectives for each scene. As a post processing step, we perform aggregation of *votes* from the model's inferred count at each angle. We experimented with three different aggregation methods: `max`, `min`, and `average`. We empirically found that `average` performed the best and improved performance over the model without aggregation.

4.4. Mismatch Loss

The original FamNet [10] architecture was trained with a single loss function:

$$density\_MSE\_loss = MSE(pred\_map, gt\_map) \quad (1)$$

The `density_MSE_loss` computes the mean squared error between the predicted and ground truth density maps.

We propose an additional loss function: `Mismatch Loss`. Intuitively, the Mismatch Losspenalizes pixels in the predicted density map which do not actually correspond to objects in the original image. As ground truth, we generate masks where a value of 1 indicates that the pixels doesn't have any objects of interest and a value of 0 indicates otherwise. The network should ideally predict a density map with 0 values in places where the mask has a value of 1. We penalize when the models predicts objects where a mask is not covering. To create masks, we convert the mask polygons from the annotation to grayscale masks.

Formally,

$$mismatch\_loss = \Sigma_{i,j}(I(i,j)) \quad (2)$$

$$I(i,j) = pred\_map[i][j] == 1 \wedge mask[i][j] == 1 \quad (3)$$

I is an indicator function which returns 1 if the `pred_map` and `mask` both have a value of 1 at the location `i,j`. If `pred_map` is equal to 1, this means the model predicted an object in that location. On the other hand, if `mask` is equal to 1, this means that the mask is white, and does not include an object. Thus, they are mismatching and should be penalized.

The *mismatch_loss* is a sum over pixels and can thus range upto the number of pixels in the entire image. To scale this appropriately, we add a hyperparamter λ to balance both the losses. We use a λ of 1^{-9} as the mismatch loss

Lastly, we combine the loss functions in equations **??** to train FamNet with a single unified loss.

$$loss = density\_MSE\_loss + \lambda * mismatch\_loss \quad (4)$$

5. Experiments

For the purpose of our project, we evaluate two baseline methods, with the first being the traditional image processing pipeline proposed in [2]. The second baseline is an object instance segmentation network (MaskRCNN) used to count the number of machined parts in the provided data set.

Drawing inspiration from [6] and some of the recent works in this domain [8, 5, 10, 13], we planned to improve the performance on the given task of machined-part counting over pre-existing networks which are generally tailored towards crowd counting tasks.

Our final technique is primarily based on the work done on learning to count objects of diverse visual categories using a few shot learning model [10] which we improved to better at our specific task using the following tricks:

1. Aggregation across different angles (averaging, min or max)
2. Additional *mismatch* loss function

5.1. Image Processing Pipeline

One of the baseline techniques implemented was to use traditional image processing techniques to obtain an estimate of the number of regularly shaped objects present in the provided data set. Given that our industry challenge task consisted of counting regularly shaped circular objects, we were able to obtain get relatively decent results using such techniques similar to the approach devised in [2]. The proposed method consisted of splitting the data set images to component channels. Using the saturation channel provided the best results for object counting. A Gaussian blu is now applied to the resulting grey scale image to remove noise.

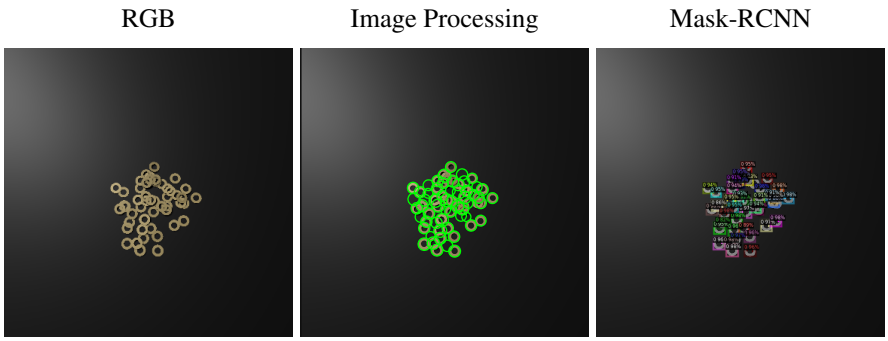

Figure 2. Baseline Predictions

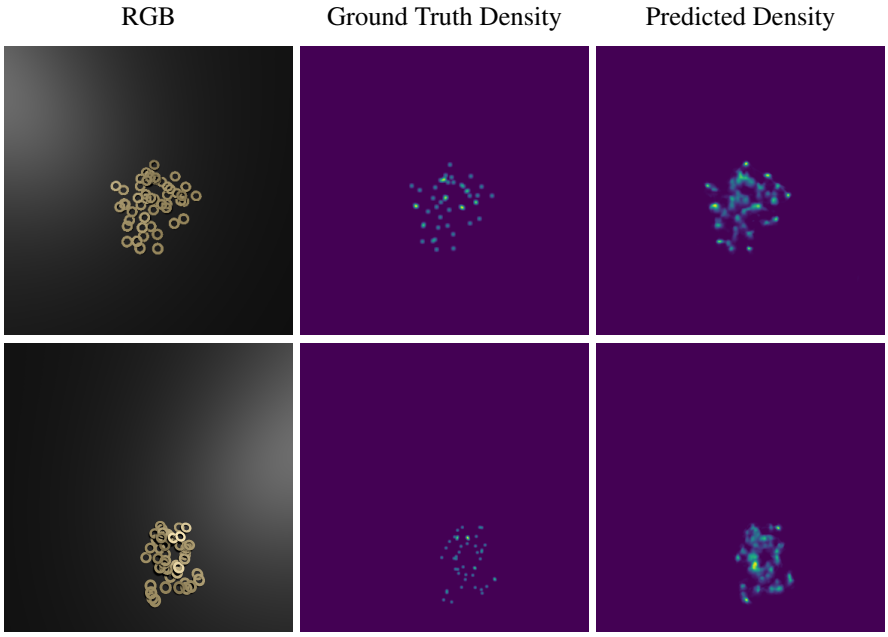

Figure 3. Predictions of the proposed method

The de-noised image is then Otsu thresholded to separate the foreground from the background. A Sobel filter is then applied to the foreground image to perform edge detection. Hough circle detection is then performed on the edge detected image to obtain a count of the number of circular objects as depicted in Figure 2. We also tuned the parameters of the Hough circles algorithm to obtain better detection performance.

5.2. Instance Segmentation (Mask-RCNN)

The first is an out of the box object detection technique, Detectron2 [14]. Object detection methods use a sliding window to identify objects locally. As a naive baseline we sum the number of object identifications in each local window to achieve a global object count. In particular we compare to the Detectron2 model: `mask_rcnn_R_50_FPN_3x`. That is, a MaskRCNN [3] with a ResNet-50 [4] and a three level feature pyramid network [7] backbone.

The original Detectron2 model was pretrained on the COCO 2017-train dataset. We include evaluation against the pretrained model as well as a finetuned model trained on the washer dataset. The performance improved from a MAE of 26.1963 to 3.7889 after finetuning. This is expected as the COCO 2017 dataset is drastically out of distribution containing large objects such as people and animals with less clutter than the washer dataset. Model predictions for the machine parts using Mask-RCNN are illustrated in Figure 2.

5.3. FamNet

To train FamNet, we minimize the proposed loss function that has density and mismatch components. We use Adam optimizer, with hyperparameters as mentioned in [10] (learning rate of 10^{-5} and batch size of 1). Every im-

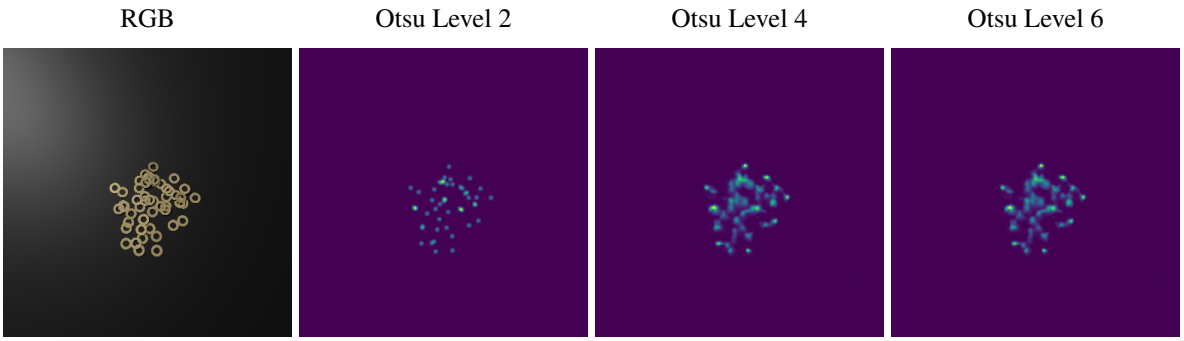

Figure 4. Multi-level Otsu Thresholding results

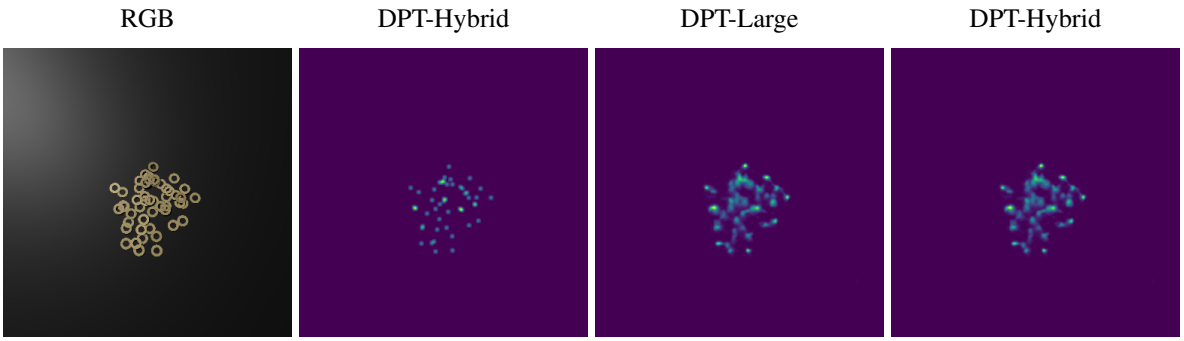

Figure 5. Depth Maps generated by DPT [9]

age is resized to 384x384 from the standard 1080x1080 image dimension in the washer dataset. We use a train-dev-test split of 80:10:10, and split images in such a way that all angles of a particular washer setup image goes in the same bucket. This is done so as to avoid any memorization of test set images that can potentially occur in a random split. We train for a total of 20 epochs for each of the experiments.

5.4. Depth-based Image Separation

Since our FamNet model already gave a very close count for input images with just a tiny MAE of 2, we noticed that the major performance drop was because of occlusions. To tackle this, we wanted to perform depth separation for our images. Our motive for the following experiments was to divide the input images into multiple depths. Once we have the depth separated images, we planned to run the FamNet model on each of the depth images individually, and take a net sum of the predictions at each level.

5.4.1 Multi Level Otsu Thresholding

To improve performance of the image processing pipeline and to use as a feature extraction technique for the main proposed method, multi level Otsu thresholding was carried out on the data set images. The Multi-Otsu threshold is a type of thresholding algorithm that is used to separate the pixels of an input image into multiple classes, similar to conventional adaptive Otsu thresholding. However, Multi-Otsu calculates several thresholds, determined by a provided count of the desired classes. The default number of classes is 3: for obtaining three classes, the algorithm returns two threshold values. By varying the number of required thresholds, Multi-Otsu thresholding can be used to segment objects in an image into different depth classes based on their grey intensity levels.

5.4.2 Dense Prediction Transformer

We implement the DPT model [9] for our problem to obtain depth maps for each image of the dataset, given its recent success. The purpose of this experiment was to separate machine parts in an image based on their depth i.e. the intensity of pixels in the depth map, so as to get a count of occluded objects better. We primarily used 3 variants of DPT: DPT-Hybrid, DPT-Large & DPT-Hybrid Finetuned on the NYU-Depth-V2 dataset. Results of this experiment are illustrated in Figure 5.

6. Evaluation and Analysis

6.1. Evaluation

We evaluate the performance of the models on the test set using the following error metrics:

Method	mae↓	rmse↓
Image Processing [2]	29.0086	37.06189
Mask-RCNN [14] (without fine-tuning)	26.1963	27.9758
Mask-RCNN [14] (with fine-tuning)	3.7889	5.0149

Table 1. Baseline Performance

Method	mae↓	rmse↓
FamNet [10]	2.60	3.64
FamNet [10] + Angle Aggregation (Min)	2.78	4.07
FamNet [10] + Angle Aggregation (Max)	3.26	3.74
FamNet [10] + Angle Aggregation (Mean)	**1.96**	**2.70**
FamNet [10] + Mismatch Loss	2.97	3.91
FamNet [10] + Mismatch Loss+ Angle Aggregation (Min)	3.06	4.35
FamNet [10] + Mismatch Loss+ Angle Aggregation (Max)	3.95	4.74
FamNet [10] + Mismatch Loss+ Angle Aggregation (Mean)	2.69	3.52

Table 2. Performance of the proposed model

- **Mean Absolute Error** (MAE): The MAE represents the average absolute difference between predicted and ground-truth values of the number of objects detected in an image.
- **Root Mean Squared Error** (RMSE): The RMSE represents the square root of the second sample moment of the differences between predicted values and observed values or the quadratic mean of these differences.

Quantitative results of the baseline methods are illustrated in Table 1, and the performance of our proposed method is illustrated in Table 2.

6.2. Analysis

In performing Multi-Otsu thresholding, it was found that due to the nature of the provided data set, objects could not be wholly separated into different depth levels via their adaptive grey scale intensity. This led to duplicate counts of objects from different thresholding levels leading to large deviations from ground truth values. Changes in lighting conditions also affected the performance of multi level Otsu thresholding with brighter areas appearing in the foreground even if they are supposed to be part of the background as shown in Figure 4 . These results disabused us of the notion that using this technique would improve network performance and it was relegated to being an unsuccessful experiment.

FamNet [10] finetuned on washers, achieves an MAE less than 2. In comparison to the Mask-RCNN and image processing techniques (3.79 and 29.01 respectively), FamNet performs well.

Through our empirical results in Table 2, we observed that the `min` aggregation has better performance than the `max` aggregation, suggesting that our technique may be over-estimating. To combat the over-estimation, we added the mismatch loss to penalize predictions in the density map where there is no object of interest in the ground truth. As one can observe from 2, the experiment with the mismatch loss increased the MAE instead. Through a closer post-mortem of the experiments, we identify that although the root cause of the minor errors were due to over counting of objects, the overestimate happens at occluded regions where there's already an object prediction. This way, the mismatch loss did not help much to fix the errors. However, we do believe that the addition of mismatch loss could help when we try to count objects of diverse scales and visual categories, unlike the ones in the washer parts dataset that has a plain black background.

7. Conclusion

In conclusion, we find that it is definitely possible to perform the task of counting machined washer parts using deep learning networks with a high degree of accuracy. The robustness of such a model to generalize to other tasks however, is called into question due to the specific nature of our training data set. Given a most general training data set with diverse visual categories, we believe that it would be possible to use this network to perform all manner of dense object counting tasks. Some avenues of future work into this area would be to try to use the different perspectives provided in the data set to generate a 3-D image of the objects. This might allow for better network performance.

References

[1] Carlos Arteta, Victor Lempitsky, J. Alison Noble, and Andrew Zisserman. Interactive object counting. In *European Conference on Computer Vision*, 2014. 2

[2] Mehmet Baygin, Mehmet Karaköse, Alisan Sarimaden, and Erhan Akin. An image processing based object counting approach for machine vision application. *CoRR*, abs/1802.05911, 2018. 1, 3, 6

[3] Kaiming He, Georgia Gkioxari, Piotr Dollár, and Ross Girshick. Mask r-cnn. In *2017 IEEE International Conference on Computer Vision (ICCV)*, pages 2980–2988, 2017. 4

[4] Kaiming He, Xiangyu Zhang, Shaoqing Ren, and Jian Sun. Deep residual learning for image recognition. *arXiv preprint arXiv:1512.03385*, 2015. 2, 4

[5] 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)*, pages 4705–4714, 2020. 1, 2, 3

[6] Victor Lempitsky and Andrew Zisserman. Learning to count objects in images. In *Advances in Neural Information Processing Systems*, 2010. 2, 3

[7] Tsung-Yi Lin, Piotr Dollár, Ross Girshick, Kaiming He, Bharath Hariharan, and Serge Belongie. Feature pyramid networks for object detection. In *2017 IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*, pages 936–944, 2017. 4

[8] Daniel Oñoro and Roberto López-Sastre. Towards perspective-free object counting with deep learning. volume 9911, 10 2016. 1, 3

[9] René Ranftl, Alexey Bochkovskiy, and Vladlen Koltun. Vision transformers for dense prediction. *ArXiv preprint*, 2021. 5

[10] Viresh Ranjan, Udbhav Sharma, Thua Nguyen, and Minh Hoai. Learning to count everything. *2021 IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR)*, pages 3393–3402, 2021. 1, 2, 3, 4, 6

[11] Dong Kyun Shin, Minhaz Uddin Ahmed, and Phil Kyu Rhee. Incremental deep learning for robust object detection in unknown cluttered environments. *IEEE Access*, 6:61748–61760, 2018. 1

[12] Vishwanath A. Sindagi and Vishal M. Patel. A survey of recent advances in cnn-based single image crowd counting and density estimation. *Pattern Recognition Letters*, 107:3–16, 2018. Video Surveillance-oriented Biometrics. 1

[13] Jia Wan, Ziquan Liu, and Antoni B. Chan. A generalized loss function for crowd counting and localization. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR)*, pages 1974–1983, June 2021. 2, 3

[14] Yuxin Wu, Alexander Kirillov, Francisco Massa, Wan-Yen Lo, and Ross Girshick. Detectron2. https://github.com/facebookresearch/detectron2, 2019. 4, 6

[15] Yifan Yang, Guorong Li, Zhe Wu, Li Su, Qingming Huang, and Nicu Sebe. Reverse perspective network for perspective-aware object counting. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR)*, June 2020. 1, 2