Density Map Estimation for Crowded Chicken

Dong Cheng, Tianze Rong, and Guitao Cao(✉)

East China Normal University, Shanghai, China
dong_cheng0525@foxmail.com, 51174500119@stu.ecnu.edu.cn,
gtcao@sei.ecnu.edu.cn

Abstract. Intensive breeding is the trend of the breeding industry. In order to make it more convenient to manage and reduce labor costs, sometimes we need to estimate the number of individuals in the poultry farm and discriminate the density distribution to help scientific management. At the same time, crowd density estimation is a developing research direction in deep learning. There are both similarities and differences between crowd counting task and chicken counting task. Aimed at the characteristics of poultry farm images, this paper presents a solution to density estimation and counting of poultry individuals in poultry farm by deep network method. We designed an end to end model and transform the problem into a pixel-level classification problem to get the density map.

Keywords: Density estimation · Flock counting · Pixel level classification

1 Introduction

Computer vision detection is playing a more and more important role in whether production or daily life. In this case, it's of imperative to estimate the density distribution of poultry in the breeding area so that makes it possible to analyze the growth of poultry. Object detection and crowd counting are two of the most relevant research directions. The method is supposed to be specialized while its requirements are also different from the ordinary detection since density counting in the chicken house is distinctive. Lack of detection of the results is not of particular concern, but the overall distribution of the flock is the focus. Specialists can infer the status of the chickens through changes in the distribution of the flock. We find in the image that it is restricted by the situation of uncertain-view, uncertain-brightness, and position-indeterminate. Usually, some chickens are incomplete within the camera lens range. Moreover, images sometimes need to be split and processed separately because of image resolution and interfering objects. It may split one chicken into two or three parts such as Fig. 1. So most

National Natural Science Foundation of China under Grant.

Y. Zhao et al. (Eds.): ICIG 2019, LNCS 11903, pp. 432–441, 2019.
https://doi.org/10.1007/978-3-030-34113-8_36

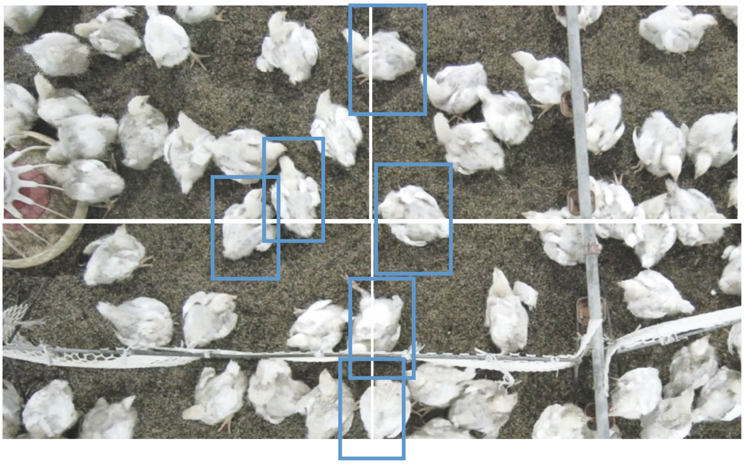

Fig. 1. As shown in the figure, some chickens are divided into multiple parts when one image is split, and the same situation exists at the edge of the image.

of the previous methods mentioned e.g. object detection does not take it into account. To effectively achieve the goal, a certain number of pictures were collected and analyzed, including daytime and night illumination. While the clarity of the images obtained is limited by the environment inside the chicken house. At the same time, it is hard to point out the head of the chicken so accurate such as crowd counting dataset. Considering that the aim of the task is not the location of a particular target, but the aggregation of the entire flock, we transfer the problem into a pixel-level problem, i.e. discriminate all pixels in the image into two classes, chicken or not. After that, we try to get the result density map by some other operations.

The paper is organized as follows. Section 2 discusses the related work. Section 3 shows the proposed technical approach followed with a description of dataset used in this paper in Sect. 4. Experiments are presented in Sects. 5 and 6 concludes this paper.

2 Related Work

Active learning is a good way to reduce the cost of manual labeling. [1] proposed a novel bidirectional active learning algorithm that explores into both unlabeled and labeled data sets simultaneously in a two-way process. [2] aimed to address the issues and develop a novel framework for effective and efficient model learning. [3] took this idea to video detection and proposed a novel weakly supervised framework. [5] used active learning to select the most informative patents for labeling.

Image segmentation is an important field in computer vision. The classic method is to use Sobel or other operators on the image. Since there are some deep learning method has come out recently, at present, the best performance at present is semantic-based image segmentation. [14] applied an up-sampling structure on image segmentation first. [21] developed the up-sampling structure before and increased the accuracy in medical image segmentation. [4] proposed a novel post-processing method based on Generative Adversarial Network is explored to reinforce spatial contiguity in the output label maps. Of course, [22] can not be ignored, it used a ROIAlign layer and mask branch to improve the accuracy of both detection and segmentation, whats more [23] is the latest development of the Mask-RCNN.

Crowd counting is the most similar task before. Most of the previous works used a multi-column architecture. [9] combined two different convolution kernels for feature extraction and obtained a good result. [10] trained a custom network with three CNN columns, each of which with a different receptive filed could capture a specific range of head sizes. However, it's high time-consuming to run three CNN columns. [12] proposed to predict which column to run for each input image patch. But the models above are all difficult to implement and they are slow at inference. To overcome the limitation, Li et al. replaced some pooling layers in the CNN with dilated convolutional filters [11,13]. Walker detect [18] in the street also combined a task of density estimation. Otherwise, [19] noticed the impact of human body structure on density estimation. [8] propose a novel feature selection based method for facial age estimation. A pre-trained model was designed to detect all kinds of parts of the human body. Whats more, [24] presented an end-to-end differentiable optical flow network for unsupervised optical flow learning. And [25] purposed a novel tube-and-droplet framework to effectively capture the rich information in 3D tube representation.

We also considered the efficiency and accuracy of the method from some other works before. [7] introduced a good method for steaming data to regress and [6] combined simultaneous sample and feature selection tasks to improve the result.

In this paper, we concluded the model structures and ideas before. It is convinced that the direction of density estimation is multi-scale image information acquisition and expansion of the receptive field. Dilated convolution [11] and multi-column [10,12] both aim at enlarging the receptive field and reducing missing pixel information, thus we considered Fully Convolutional Networks(FCN) [14], replacing the last few fully connected layers by convolutional layers to make efficient end-to-end learning and inference that can take arbitrary input size. Its up-sampling operation is a useful way to expanse the receptive field. We combined the advantage of FCN and designed our own model to finish this work, and then get the density map by a density estimation with the feature map that Network output.

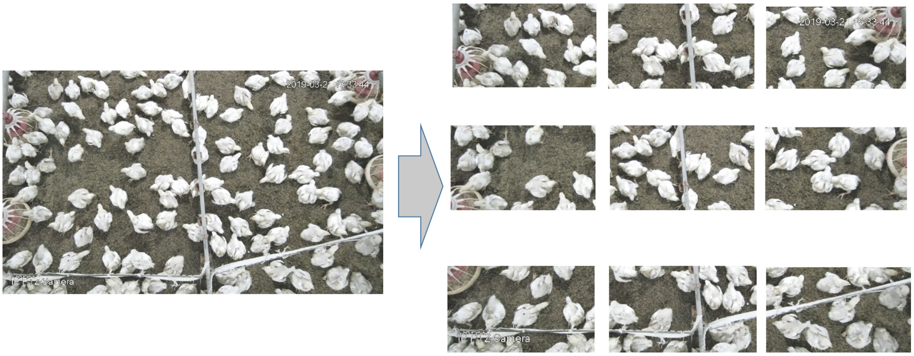

Fig. 2. The images we collected are shown. And when we split the source images, we may take the object apart, which influence the result a lot.

3 The Proposed Approach

3.1 Pixel Level Classification

Compared to general target detection, the process of density estimation needs to expand the receptive field in the process of extracting features. Like the dilate convolution in [13] and so on. Instead of traditional object counting work, we try to point each pixel of the object to avoid complex conditional constraints on detection. So we choose FCN [14] structure to achieve this goal. This net is convenient and converts our work into a binary classification problem. Of course, we also regress some other parameters in a pixel-level classification to finish the estimation task. We named it ChickenNet, which is an end to end model and the size of the input image is arbitrary.

3.2 Net Design

Fully connected network is widely used in image segmentation. Unlike other density estimation tasks, we have some particular difficulties. First of all, there is no such amount of images to train, and the crowd counting datasets are also invalid in this task. Whatsmore, the training set of the flock is difficult to calibrate. So we try another way that regardless of the object as a point for density estimation. We designed a network based on FCN to discriminate all pixels on the graph into two categories. And then use the output feature map to get the result. Figure 3 is the main architecture of the network. We choose VGG16 [15] to extract image features, imitate the FCN, remove the last fully connected layer and then perform the up-sampling operation. We can get the result of pixels' classification, which likes a mask of the source image. Then calculate a Gaussian kernel to do Gaussian blur operation, and obtain the final density map. In order to regress the size of the chicken, we add a new parameter S, and then change the loss function for it.

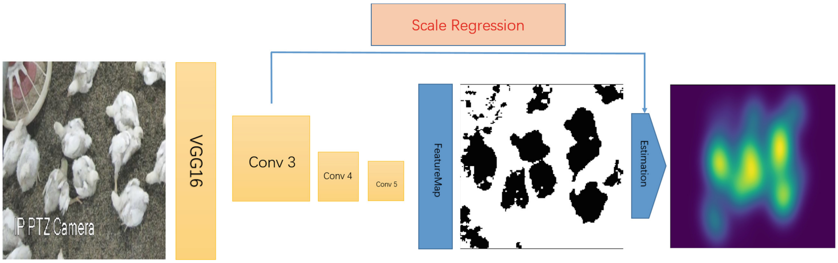

Fig. 3. The image data flow is shown. We get the feature map from our network, and then a special estimation module is designed to generate the Density map. The brighter the area, the higher the density.

3.3 Chicken Scale Estimation

Chicken scale estimation is a very important part in our work. We don't regard the objection as a point differs from other estimation methods, which means we need some different methods to generate the density map. Through a convolutional neural network, we can get the feature map of the images, it helps to determine if there is a target at the pixel location. Of course, we are supposed to have another way to transform it to a density map. In order to estimate the distribution of the chicken, it's essential to know the size of the chicken from the collected images. Therefore, in our ChickenNet model, before up-sampling, we add a fully connected layer to regress the average number of the pixels of each chicken, and the loss function is also changed. For decreasing the influence of the classification task, we define the loss function as following. M is a constant, we set M is 200 to get the best density map.

$$loss = BCEloss + \|S - PS\|_2 /M \tag{1}$$

3.4 Density Map Generating

According to the output feature map, we need to generate a density map quickly and accurately. Kernel density estimation [16] is applied in many other methods of density estimation to get the final heatmap. We first used this method. But, the density of the entire image needs to be updated when calculating the Gaussian kernel for each point, but our output image has too many points to calculate. So it costs too much to generate the density map. However, to some extent, our method has got a mask of all object chickens, which reflected a certain distribution. By calculating a Gaussian kernel and execute Gaussian blur operation, the results are remarkable in our data set. As is shown in Fig. 3.

Gaussian blur is a very simple and common image processing method. Sometimes it may surprise you. This is the process of resetting each pixel value in the image by setting each pixel to the average of the surrounding pixels. And normal distribution is obviously a weight distribution model. Two-dimensional Gaussian function is shown.

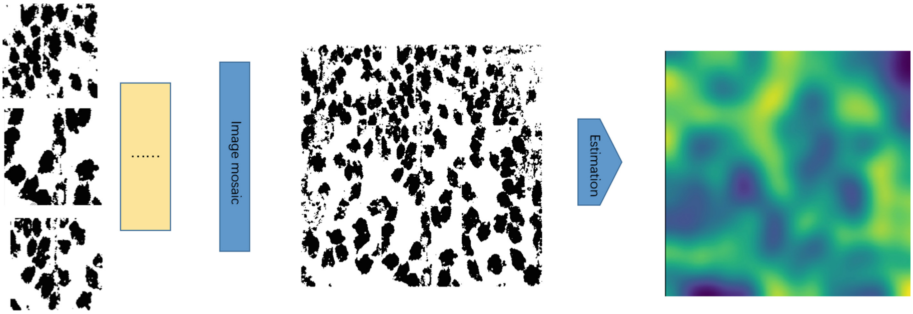

Fig. 4. The idea of segmentation while generating a density map is shown. It can be used to improve detection results.

$$G(x,y)=\frac{1}{2\pi\delta^2}e^{(-x^2+y^2)/2\delta^2} \tag{2}$$

And then we need to set the value of δ to calculate the weight matrix. Each point is multiplied by its own weight value, and after accumulating, the Gaussian blur result of the center point is obtained. And in our work, this simple method can help to transform the mask of the source image to a density heat map. The only one should be considered is the size of the Gaussian kernel. To some extent, the size of the object chicken.

4 Dataset

We collected about 100 images from different view angles in some chicken houses. And for these images, it's difficult to point out the chicken head. So we choose other ideas instead of head counting estimation. Before importing the images to the purposed approach, we need to preprocess the collected pictures to improve the accuracy of the estimation. We choose about half of the collected images to make our train set. A mask image for each image in the training set and estimate the average size of the target. For convenience, we assume that the value of the target size is represented by a square box, so one parameter S is enough. As is shown in Fig. 5.

4.1 Image Splitting

The images that we collected are too big to detect and to estimate. In order to reduce noise pixels and the number of pixels that need to be processed. We divide one image into nine parts, just like Fig. 2, then detect chickens in each part. Thanks to the idea of segmentation, it's unnecessary to care if one chicken has been divided into several parts. Because we deal with the problem at pixel-level, and each pixel will only be divided into two classes. Finally, we contact the nine output feature map in order and transform them into a big density map. As is shown in Fig. 4.

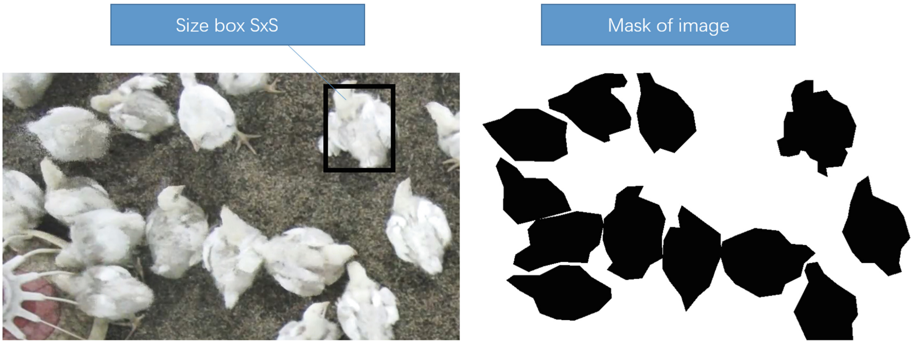

Fig. 5. The train dataset that we made for the model. Including source images, mask images, and their average object size S.

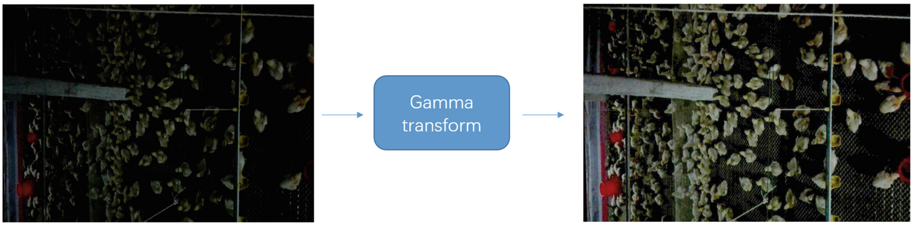

Fig. 6. Use gamma transform to process lower brightness pictures.

4.2 Gamma Luminance Transformation

In the dataset, some images are in a dark situation. So gamma transform is considered. It is shown in Fig. 6. Gamma transform can enhance the contrast of the image, and the basic form of gamma transformation is

$$s = cr^{\gamma} \tag{3}$$

r is the gray value of the input image, s is the gray value of the output image, and c is just a constant. γ is the variable needs to adjust. After some trials, we choose a parameter γ for our collected images.

4.3 Ground Truth

We identified all the locations of the pixels that contain the target, and get a pixel point set as ground truth(GT). In other words, we define some areas of the source image as ground truth. In order to compare this method to others. We count the number of missing and misclassified pixels as an evaluation indicator for the model. We set ChickenNet to predict pixel set as PT. Considering our model outputs more points, we decide to calculate two values as follows. The smaller the values are, the better the effect of the model.

$$P1 = count(PT \setminus GT)/count(GT) \tag{4}$$

Table 1. Comparing two methods

Methods	YoloV3	ChickenNet
P1	16%	1.3%
P2	9%	2.5%

$$P2 = count(GT \setminus PT)/count(GT) \tag{5}$$

5 Experiments

The experiment involves an object detect method (yolov3) [17] and our purposed model. In order to reduce the probability of missed detection, the source images are split into nine parts. And then we compare these two methods by P1 and P2. The scenes of the captured images have similarities, so about ten images is enough to train the detection model. We use about 50 images to test these two methods, and their P1, P2 are shown in Table 1.

5.1 Object Detect Method

We train a yolov3 object detection model to deal with the data. The yolov3 model is one of the best object detection models. And we have also tried other models like yolov2 and faster-RCNN [20], but they all do not work. While training the model, only several train images are required. We make the train set and doesn't box the location of the target where the edge portion is cut. But after detection, still some 'chicken parts' are detected as a complete target. This may influence the final density map result. For example, the edge of the image being cut will have more of the detected object, making it brighter in the density map. After detection, kernel density estimation is considered to generate the density map. The detection output is replaced with a two-dimensional Gaussian kernel at each coordinate point, then superimpose all the Gaussian kernel to get the heat map. As is shown in Fig. 7, all detected locations are identified. And according to the detection result, there are still some missed chickens. Meanwhile, it takes about 3 min for one image to generate its density map, it is too long if we want real-time monitoring.

5.2 ChickenNet Method

The train set has 30 images. Including source images, mask images, and their average object size S. Although the output may result in a little more disturbing pixels, it does not influence the whole distribution; On the other hand, the output density map is still accurate. And the model can be trained quickly. So we can get the conclusion that our method has a higher fault tolerance rate, and can truly reflect the aggregation of chickens for this situation. Meanwhile, our model is easier to train. So it can update itself quicker and easier than other models.

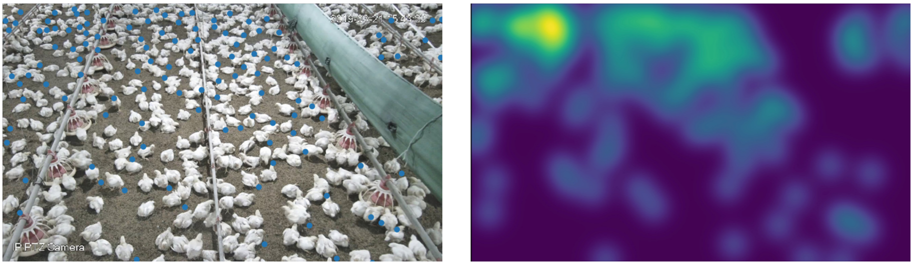

Fig. 7. Yolo v3 detection result and the density map generated by kernel density estimation.

6 Conclusion

In this paper, we present a new approach based on a fully connected network for crowded chicken density estimation and name the new model ChickenNet. We transform the problem into a pixel level problem, and aiming to our image set, we find Gaussian blur idea can simply generate the density map. An end to end model is completed at last. However, we do the other experiment to compare. Use yolo detection model and kernel density estimation to do chicken density estimation task. The result is acceptable but not as good as our proposed model.

References

1. Zhang, X.Y., Wang, S., Yun, X.: Bidirectional active learning: a two-way exploration into unlabeled and labeled data set. IEEE Trans. Neural Netw. Learn. Syst. **26**(12), 3034–3044 (2015)
2. Zhang, X.Y., Shi, H., Zhu, X., et al.: Active semi-supervised learning based on self-expressive correlation with generative adversarial networks. Neurocomputing **345**, 103–113 (2019)
3. Zhang, X.Y., Shi, H., Li, C., et al.: Learning transferable self-attentive representations for action recognition in untrimmed videos with weak supervision. arXiv preprint arXiv:1902.07370 (2019)
4. Zhu, X., Zhang, X., Zhang, X.Y., et al.: A novel framework for semantic segmentation with generative adversarial network. J. Vis. Commun. Image Represent. **58**, 532–543 (2019)
5. Zhang, X.: Interactive patent classification based on multi-classifier fusion and active learning. Neurocomputing **127**, 200–205 (2014)
6. Li, C., Wang, X., Dong, W., et al.: Joint active learning with feature selection via CUR matrix decomposition. IEEE Trans. Pattern Anal. Mach. Intell. **41**(6), 1382–1396 (2018)
7. Li, C., Wei, F., Dong, W., et al.: Dynamic structure embedded online multiple-output regression for streaming data. IEEE Trans. Pattern Anal. Mach. Intell. **41**(2), 323–336 (2018)
8. Li, C., Liu, Q., Dong, W., et al.: Human age estimation based on locality and ordinal information. IEEE Trans. Cybern. **45**(11), 2522–2534 (2014)

9. Boominathan, L., Kruthiventi, S.S.S., Babu, R.V.: CrowdNet: A Deep Convolutional Network for Dense Crowd Counting (2016)
10. Zhang, Y., Zhou, D., Chen, S., et al.: Single-image crowd counting via multi-column convolutional neural network. In: Computer Vision & Pattern Recognition (2016)
11. Chen, L.C., Papandreou, G., Kokkinos, I., et al.: DeepLab: semantic image segmentation with deep convolutional nets, atrous convolution, and fully connected CRFs. IEEE Trans. Pattern Anal. Mach. Intell. **40**(4), 834–848 (2018)
12. Sam, D.B., Surya, S., Babu, R.V.: Switching Convolutional Neural Network for Crowd Counting (2017)
13. Li, Y., Zhang, X., Chen, D.: CSRNet: Dilated Convolutional Neural Networks for Understanding the Highly Congested Scenes (2018)
14. Long, J., Shelhamer, E., Darrell, T.: Fully convolutional networks for semantic segmentation. IEEE Trans. Pattern Anal. Mach. Intell. **39**(4), 640–651 (2014)
15. Simonyan, K., Zisserman, A.: Very deep convolutional networks for large-scale image recognition. Computer Science (2014)
16. Härdle, W.: Kernel Density Estimation. Smoothing Techniques. Springer, New York (1991)
17. Redmon, J., Farhadi, A.: Yolov3: an incremental improvement. arXiv preprint arXiv:1804.02767 (2018)
18. Zhang, N.C., Li, N.H., Wang, X., et al.: Cross-scene crowd counting via deep convolutional neural networks. In: 2015 IEEE Conference on Computer Vision and Pattern Recognition (CVPR). IEEE Computer Society (2015)
19. Huang, S., Li, X., Zhang, Z., et al.: Body structure aware deep crowd counting. IEEE Trans. Image Process. **27**(3), 1049–1059 (2017)
20. Ren, S., He, K., Girshick, R., et al.: Faster R-CNN: towards real-time object detection with region proposal networks. In: International Conference on Neural Information Processing Systems (2015)
21. Ronneberger, O., Fischer, P., Brox, T.: U-net: convolutional networks for biomedical image segmentation. In: Navab, N., Hornegger, J., Wells, W.M., Frangi, A.F. (eds.) MICCAI 2015. LNCS, vol. 9351, pp. 234–241. Springer, Cham (2015). https://doi.org/10.1007/978-3-319-24574-4_28
22. He, K., Gkioxari, G., Dollár, P., et al.: Mask r-cnn. In: Proceedings of the IEEE International Conference on Computer Vision, pp. 2961–2969 (2017)
23. Liu, S., Qi, L., Qin, H., et al.: Path Aggregation Network for Instance Segmentation (2018)
24. Ren, Z., Yan, J., Ni, B., et al.: Unsupervised deep learning for optical flow estimation. In: Thirty-First AAAI Conference on Artificial Intelligence (2017)
25. Lin, W., Zhou, Y., Xu, H., et al.: A tube-and-droplet-based approach for representing and analyzing motion trajectories. IEEE Trans. Pattern Anal. Mach. Intell. **39**(8), 1489–1503 (2017)