Automatic counting of chickens in confined area using the LCFCN algorithm

Diab Abuaiadah
Centre of IT
Waikato Institute of Technology
Hamilton, New Zealand
Diab.Abuaiadah@Wintec.ac.il

Alexander Switzer
Information Services Department
Gallagher Group Limited
Hamilton, New Zealand
alex@alexswitzer.com

Michael Bosu
Centre of IT
Waikato Institute of Technology
Hamilton, new Zealand
Michael.Bosu@Wintec.ac.nz

Yun Liu
Development team
More4Apps
Hamilton, New Zealand
lily.liu@more4apps.com

***Abstract*— Grouping chickens based on their weights is an important process that takes place in many chicken farms in New Zealand where chickens are grouped into three categories: small, medium and large. Each category has pins (cages) to temporarily hold the chickens during the process and a permeant bigger section to hold the chickens after grouping. Chickens are weighed and placed in respective pins. Thereafter they are released to the permanent section. Currently, the chickens are counted manually when they are released from a pin to a bigger section. The task of weighing chickens, placing them in a pin and releasing them to a bigger section is repeated until all chickens are moved to their respective bigger section and the total number of chickens in each section is calculated. This manual effort is done by several employees and takes several hours. This study investigated the feasibility of using deep learning algorithms to replace the manual counting. We applied the localized fully convolutional network (LCFCN) algorithm to count and locate chickens from images of the pins. LCFCN was applied to a dataset of 4092 images containing 114132 chickens. The algorithm was evaluated using the Mean Absolute Error (MAE), Mean Absolute Percentage Error (MAPE) and Root Mean Square Error (RMSE) metrics and achieved the values of 0.5592, 1.36% and 1.67 respectively which are promising results in this setting. Furthermore, we modified the implementation of LCFCN to enable a user to manually alter the predicted labels to guarantee error free counting and localization.**

Keywords—deep learning, LCFCN, convolutional neural networks, object counting, point-level annotation, image processing

I. Introduction

Recent advancements in computation power, graphic processing unit, artificial intelligent and neural networks led to the development of several efficient algorithms and architectures. Deep learning architectures got the attention of many academic researchers and industry experts who steered the development of several algorithms and platforms [1].

Computer vision is an important subject and has numerous numbers of real-life applications. Deep learning architecture advanced this field and presented several new algorithms including image classification, colorization, object recognition and object counting [2]. This paper focuses on object counting which has many real-life applications such as crop assessments, animal counting, underwater fish counting and tree counting [3].

Labelling objects for training is a major task and requires massive human effort. In many cases, labelling one object using bounding box or semantic segmentation techniques can take several seconds and this is a challenge for large datasets or when images contain a large number of objects. The task of simple counting may not necessitate this type of elaborative labelling. Subsequently, the concept of weak supervision, dotting or point-level labelling (annotation) has been introduced [3].

Point-level annotation requires only one click for each object (which result in marking one pixel). Obviously, this technique is significantly quicker than other labelling techniques and the level of human effort and expertise for labelling is dramatically reduced. This type of labelling proved to be effective for object counting and the associated algorithms are competitive or outperform the state-of-the-art algorithms for strong supervision such as bounding boxes [4], [5].

LeCun et al. introduced Convolutional neural networks (CNN) [6]. This architecture is a type of deep learning architecture which has a kernel that filters the data to capture spatial and temporal dependencies. In recent years, extensive experiments showed a clear superiority of this architecture in many applications, mainly image processing [7].

Fully convolutional network (FCN) is a CNN without the dense layers which means it only contains convolutional layers [8]. The absence of dense layers makes the network efficient and able to handle inputs with different sizes.

Laradji et. al. [4] introduced a new approach for object counting which doesn't consider the sizes and shapes of objects and only requires point-level annotation. This interesting approach outperforms many state-of-the-art algorithms that uses stronger labelling such as depth features, multi-points and bounding box labelling in many settings. They introduced a new novel loss function which leads the neural network to output a single blob per object. The loss function is denoted as *localization-based counting loss* (LC). This technique is based on fully convolutional neural network (FCN) and, subsequently, the new approach is referred to as LC-FCN or LCFCN.

For chicken farmers, grouping chickens is a complex and expensive process. Currently it requires human effort for weighing and counting. This study showed the feasibility of automating the counting task using LCFCN algorithm. This proof of concept has the potential to reduce human effort and improve the accuracies for many chicken farms in New Zealand and worldwide. The experiments of LCFCN produced perfect counting for 72% of the images and the number of errors in the remaining images is minimal. Furthermore, we modified the implementation of LCFCN to enable a user to alter the predicted results of the algorithm to guarantee error free counting.

The manuscript is organized as follow: Section II presents the literature review. Section III describes the methodology which includes metrics for evaluation, dataset and settings. Section IV reports the results and discusses the factors that influence the algorithm. Section V provides conclusion and future work.

II. RELATED WORK

As stated in the introduction, object counting in computer vision has many real-life applications. Most of recent algorithms are based on point-level annotation and CNN. This section augments the introduction and highlights relevant and recent publications.

Zhang et. al. [9] introduced a method for crowd counting on metro platforms based on CNN architecture. This method was applied to a dataset of 627 images which contains 9243 annotated heads.

Saleh et. al. [10] built large scale datasets for underwater fish-habitat where state-of-the-art deep learning algorithms were trained and tested. It was shown that these algorithms are effective and has the potential to produce satisfactory results when counting the number of fish from images and other classification tasks. They also showed the impact of pretraining on ImageNet to the performance of these algorithms. The experiments used point-level and per-pixel annotations.

Laradji et. al. [11] applied two deep learning algorithms to count the number of cows from satellite images. The two algorithms are: CSRNet which is a density-based for counting and localization of objects and LCFCN. Both methods require point-level labelling. Their experiments showed the effectiveness of these two architectures but also the need for further investigations. They also stated that the resolution of images can be a decisive factor on the performance of these algorithms.

Tian et. al. [12] used a combination of counting convolutional neural network and ResNeXt architecture to count the number of pigs in an image and achieved 1.67 Mean Absolute Error per image. They stated that this result is better than competing algorithms.

Cheang et. al. [13] used convolutional neural network to count palm trees from satellite images with resolution range from 40cm per pixel to 1.5 meter per pixel. The network was trained with 500 images where each image was cropped to 40 pixels X 40 pixels. Their experiments showed above 99% accuracy.

Santos de Arruda et al. [14] used CNN architecture to count and locate high density objects. They applied their implementation to a car and tree counting datasets. They used mean absolute error and root-mean-squared error to evaluate the performance of their implementation. Their implementation showed superior results compared to that of the state-of-the-art algorithms.

Saleh et. al. [15] proposed a segmentation model which is based on point-level supervision to estimate fish body measurements (length, width and mass). They used a fully CNN with one random walk to get per-pixel segmentation. The model used LCFCN loss function. Their experiments showed that this model outperforms fully supervised state-of-the-art algorithms in many settings.

Robinson et. al. [16] applied the LCFCN algorithm to count and detect cows and elks from a very high-resolution satellite imagery. Using precision and recalls metrics, they reported that LCFCN outperformed other competitive counting algorithms and in three tests scenes.

III. METHODOLOGY AND DATASET

A. Metrics for Evaluation

The literature includes several metrics to evaluate counting algorithms. Consider the set $T = \{t_0, \ldots, t_{N-1}\}$ of images which is used for testing. The absolute error for an image t_i is defined as $|e_i - g_i|$, where e_i is the total number of objected marked (predicted) by the algorithm and g_i is the total number of objects marked by the ground truth. The mean average error (MAE) measurement [17] is defined as in equation 1.

$$\frac{1}{N}\sum_{i=0}^{N-1}|e_i - g_i| \quad (1)$$

This measure is oblivious to object locations. Suppose the algorithm marks a location that doesn't correspond to an object and fails to mark an existing object. In this scenario, these two errors cancel each other, and the errors will not be factored in this evaluation. Another drawback of this metric is the failure to consider the total number of objects in an image. Consider a counting algorithm predicts 98 out of 100 objects in an image and another counting algorithm predicts three out of five objects. This metric allocates the same score to both algorithms. It is reasonable to assume that the first algorithm produced better results than the second algorithm.

The Grid Average Mean Absolute Error (GAME) measurement aims to tackle the drawback of localization in MAE [17]. The grid (image) is split into 4^L equal and non-overlapping regions, where $, L\ is\ an\ integer\ \geq 0$ and MAE is used to evaluate each region in the image. When setting L=0, the definition of GAME is equal to that of MAE. If there are many objects crossing different regions, this distort the evaluation and increasing the value of L can lead to a significant distortion in many scenarios, mainly images with a high number of objects.

MAPE is a variation of MAE that considers the total number of objects. It divides the absolute error by the total number of objects in an image and average these values over all the images in the testing set. Formally, it is calculated as in equation 2.

$$\frac{1}{N}\sum_{i=0}^{N-1}\frac{|e_i - g_i|}{\max(g_i, 1)} \quad (2)$$

RMSE captures the dispersion in the differences between the predicted and actual count and it is calculated as in equation 3.

$$\sqrt{\frac{1}{N}\sum_{i=0}^{N-1}|e_i - g_i|^2} \quad (3)$$

The experiments in this study reports the values of MAE, MAPE and RMSE measures. In addition to these measures, the total number of manual alterations to guarantee error free counting and localization is considered. The total number of manual alterations is an indication of the quality of counting and localization.

B. Building and Labeling the Datset

The grouping process takes several hours where the total number of chickens involved in this process can reach several thousands. Since chickens are continually added to pins and removed from pins, the total number of chickens in any given pin is frequently changing.

During a real-life grouping in a chicken farm in New Zealand - we are unable to disclose the name of the organization due to confidentiality - we took 700 images using an iPhone device and a digital camera. Images were captured from different angles and positions with target pins containing different number of chickens. The grouping process took place in a closed structure where lighting was adjusted for the well-being of the chickens. Subsequently, the lighting was low (poor) which affected the quality of images and consequently, many had poor quality (blurry) resulting in those images being discarded from the dataset.

In subsequence to poor lighting, another challenge was labelling crowded and overlapping objects in an image. Dotting (using the labelling tool) the objects is not a straightforward task and the quality of labelling is unclear. Images with higher resolution can improve the quality of labelling and the overall quality of the dataset.

From the 700 images, 341 images were retained and used to form the core of the dataset. These images contained pins with different numbers of chickens. The size of each image is approximately 4MB where each image covers an approximate area of 100cm x 100cm (the area of a pin).

The images in the core dataset were cropped applying commonly used techniques [15] to 1000pixels x 1000pixels and stored as PNG format. The dataset was labelled using an in-house customised labelling tool. The labelling tool displayed the total number of chickens and provided the user with the adding and removing label functionalities. A user can add a label (dot) that corresponds to a chicken (object). The added label is translated to a single pixel where the coordinate of the pixel is then added to the mask text file. Similarly, the user also can unclick an object which results in removing the object (the dot) from the image and removing the corresponding coordinate from the mask text file.

For expanding the dataset, we created 12 versions of each image. These versions consisted of four rotated images (each image was rotated 90 degrees and captured) with each rotated image then being mirrored and flipped. Subsequently, the total number of images in the dataset is $341x12 = 4092$. The total number of chickens (objects) was 114132 where the average number in each image was approximately 28 chickens. The standard deviation of the number of chickens in the dataset is 21.99.

Fig. 1 illustrates the distribution of the number of chickens in the dataset and Fig. 2 presents an example of a labelled image using our in-house customized labelling tool.

C. Experiments and Implementation

In chicken farms in New Zealand, the grouping process has few slightly different setups, but all include pins to temporarily hold chickens belonging to a group (category) based on weight. The chickens are then released to a respective larger and permanent section. The *manual process of counting* occurs when the chickens are moved from the pins to the respective bigger sections. The task of weighing chickens and putting them into pins and then to bigger and permanent sections is repeated until all chicken are graded. A pin has the capacity to temporarily hold a small number of chickens (less than 200) and a bigger section has the capacity to contain several thousands of chickens. Our experiments used LCFCN to count the number of chickens in images of the pins to replace the manual counting.

The implementation is based on the code of LCFCN which is free and publicly available for downloading from the GitHub repository [10]. The implementation includes a text file (mask) which holds the labels of the chicken objects. Considering that all images are cropped to a 1000pixels x 1000pixles, the text file contains rows of items, of which each row is an x and y coordinate of one chicken object. Therefore, the total number of rows is equal to the total number of chickens. Furthermore, each coordinate represents the location of the chicken in the image.

Our in-house labelling tool accesses the text file and draws dots (see Fig. 2) on the images to represent the objects. The dots represent the coordinates (locations) of the objects in an image.

Given these technical details, a user can manipulate the labelling tool to guarantee *free of error counting* as the user can manually add and remove dots to any image. A user can click on any part of an image to add a new dot (object) or can click an existing dot to remove it. By adding dots, the location of the dot is translated to adding the respective coordinate in the text file. Respectively, clicking an existing dot in the image, results in removing the label from the image and the coordinate from the text file. In addition to this, the labelling tool displays the total number of objects (chickens) in the image and updates the dots on the display.

When applying the LCFCN algorithm, it produces the mask (text file) as the predicted labels. This file is used as an input to our labelling tool. A user can manually review and adjust the results of LCFCN algorithm.

The original implementation of LCFCN used Python language and was executed under the Linux operating system. This was manipulated and executed under the Windows operating system with eight cores, 16 GB of RAM and a NVIDIA GTX 1060 GPU card.

The experiments applied the common settings used by many researchers. It randomly split the dataset to 70% for training, 10% for validation and 20% for testing. The number of images in the testing set is 819 containing 23335 chickens in total. For pretraining, we used the ResNet50 transfer learning model.

The experiments were set to run LCFCN for 100 epochs. The values of the loss function were reported after each epoch. At the end, after the 100 epochs, the values of MAE, MAPE and RSME were calculated. In addition to this, a manual manipulation is provided to guarantee a free of errors counting for the 819 images in the testing set. This measure provides an indication of the quality of localization of the LCFCN algorithm.

Bin	Frequency
10	720
20	1068
30	960
40	672
50	264
60	72
70	60
80	24
90	60
100	120
More	72

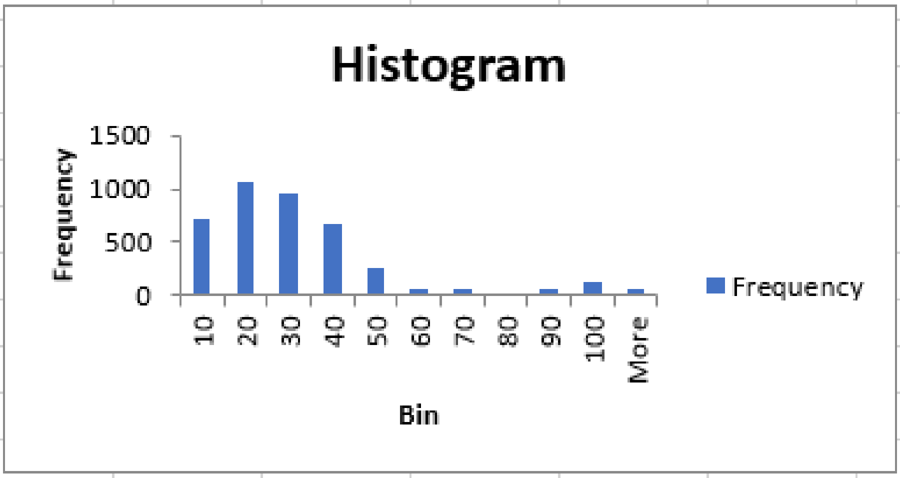

Fig. 1: basic information of the dataset.

Fig. 2: an example of labelled image in a pin.

IV. RESULTS AND DISCUSSIONS

The experiments captured the values of the loss function after each epoch, the values of MAE, MAPE and RSME after 100 epochs. Furthermore, a manual effort has been applied to correct the predicted labels of LCFCN and to estimate the human effort to guarantee free of errors counting.

Fig. 3 presents the values of the loss function after each epoch. The values after each of the first three epochs are approximately 160, 8 and 3. This shows the impact of pretraining the network on ResNet50 model. After the first three epochs, the value of loss function fluctuates and gradually decrease but remained above the value one.

For MAE and MAPE. The LCFCN algorithm achieved 0.5592 and 1.36% respectively. The average number of chickens in an image in the testing set is $\frac{23335}{819} = 28.49$. LCFCN produced perfect counting for 590 images, over counting (predicted labels are more than the ground truth) for 168 images and under counting (predicted labels are less than the ground truth) for 61 images. The total number of chickens in the over counted images is 338 and in the undercounted images is 120 (in 61 images). This result indicates that LCFCN predicts accurate number of chickens for most images and when there is an error, there are minimal number of labels to be corrected. Although perfect counting may not indicate perfect localization this is an encouraging result and it highlights the potential of LCFCN.

The reported value of RSME is 1.6. This small value indicates that LCFCN performs well when the number of objects (chickens) in the images is relatively high. There are several issues with such images, and this will be further discussed in the conclusion and future research section.

Fig. 4, Fig. 5 and Fig. 6 present examples of the output of LCFCN and the required manual alterations to manipulate the result to attain free of errors labelling. For these figures (a) represents the ground truth and (b) represents the predicted labels of LCFCN. The labelling tool gathers meta data and present it on the image to facilitate the manual alterations. Fig. 4 shows an example where LCFCN achieved correct labelling for an image that contain 23 chickens. In this scenario, no alterations are required. Fig. 5 shows an example where LCFCN labelled a chicken twice. The image contains 45 chickens and the black circle shows the wrong labelling of the chicken. The user can remove the additional label to align the prediction of LCFCN to the ground truth. For this image, one alteration is required to achieve perfect labelling. Fig. 6 shows another type of wrong labelling by LCFCN where the black circle shows a label of LCFCN that has no respective object. The number of objects in this image is 30. A user can manually remove this label using a single click which means only one alteration is required to achieve perfect labelling for this image.

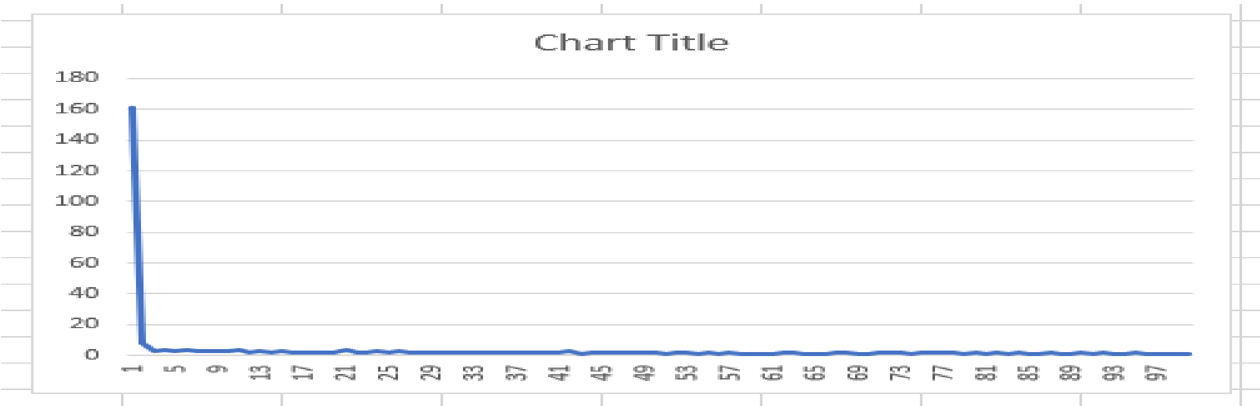

Fig. 3: the values of loss function after each epoch.

Fig. 4: the ground truth vs the predicted labels where no alterations are required.

Fig. 5: LCFCN labelled a chicken twice.

(a) (b)

Fig. 6: LCFCN labelled an empty spot as a chicken.

V. Conclusion and Future Research

Point level labelling is a recent and promising technique which has many applications. Unfortunately, not many datasets are publicly available. We are in the process of obtaining more images with higher resolutions and better qualities. In addition to this, we are aiming to obtain the consent of the organization to make these images with our labelling publicly available.

LCFCN produced excellent counting results for images which contain low number of chickens. For crowded images where the number of chickens reached 100 and there is a high degree of overlapping, LCFCN produced good counting results. However, for such images, labelling becomes a challenge and the ground truth is unclear. In these scenarios, the display is poor to human eyes and it is a challenge to establish the ground truth and to evaluate the algorithm. To draw a more robust conclusion, it is preferable to acquire images with higher resolution that contain a high number of chickens.

Evaluating the quality of localization for this dataset is a challenge. The predicted labels of LCFCN for images with low numbers are excellent. In such cases no more than one alteration is required to manually correct the result of LCFCN.

We are planning to compare the performance of LCFCN with CSRNet and ResNext architectures once images with higher resolution are available.

VI. References

[1] A. Reuther, P. Michaleas, M. Jones, V. Gadepally, S. Samsi and J. Kepner, "Survey and benchmarking of machine learning accelerators," in *IEEE high performance extreme computing conference (HPEC)*, 2019.

[2] M. Hassaballah and A. I. Awad, Deep learning in computer vision: principles and applications., CRC Press, 2020.

[3] V. Lempitsky and A. Zisserman, "Learning to count objects in images," *Advances in neural information processing systems,* vol. 23, pp. 1324-1332, 2010.

[4] I. Laradji, N. Rostamzadeh, P. Pinheiro, D. Vazquez and M. Schmidt, "Where are the blobs: Counting by localization with point supervision," in *In Proceedings of the European Conference on Computer Vision (ECCV)*, 2018.

[5] D. Onoro-Rubio and R. López-Sastre, "Towards perspective-free object counting with deep learning," in *In European conference on computer vision*, 2016.

[6] Y. LeCun, B. Bernhard, J. S. Denker and D. Henderson, "Backpropagation applied to handwritten zip code recognition," *Neural computation,* vol. 1, no. 4, pp. 541-551, 1989.

[7] A. Khan, S. Anabia, U. Zahoora and A. Qureshi, "A survey of the recent architectures of deep convolutional neural networks," *Artificial Intelligence Review,* vol. 53, no. 8, pp. 5455-5516, 2020.

[8] J. Long, E. Shelhamer and T. Darrell, "Fully convolutional networks for semantic segmentation," in *In Proceedings of the IEEE conference on computer vision and pattern recognition*, 2015.

[9] J. Zhang, J. Liu and Z. Wang, "Convolutional Neural Network for Crowd Counting on Metro Platforms," *Symmetry ,* vol. 13, no. 4, 2021.

[10] A. Saleh, I. Laradji, D. Konovalov, M. Bradley, D. Vazquez and M. Sheaves, "A realistic fish-habitat dataset to evaluate algorithms for underwater visual analysis," *Scientific Reports,* vol. 10, no. 1, pp. 1-10, 2020.

[11] I. Laradji, R. Pau, F. Kalaitzis, D. Vazquez, R. Young, E. Davey and A. Lacoste, "Counting Cows: Tracking Illegal Cattle Ranching From High-Resolution Satellite Imagery," in *arXiv preprint arXiv:2011.07369*, 2020.

[12] M. Tian, H. Guo, H. Chen, Q. Wang, C. Long and Y. Ma, "Automated pig counting using deep

learning," *Computers and Electronics in Agriculture,* vol. 163, no. 104840, 2019.

[13] E. K. Cheang, T. K. Cheang and Y. H. Tay, "Using convolutional neural networks to count palm trees in satellite images," in *arXiv preprint arXiv:1701.06462*, 2017.

[14] M. Dos Santos de Arruda, O. Lucas Prado, A. Plabiany Rodrigo, G. Diogo Nunes, J. Marcato Junior, A. Paula Marques Ramos, E. Takashi Matsubara, Z. Luo, J. Li, J. de Andrade Silva and W. Nunes Gonçalves, "Counting and Locating High-Density Objects Using Convolutional Neural Network," *arXiv e-prints, arXiv-2102,* 2021.

[15] A. Saleh, I. Laradji, P. Rodriguez, D. Nowrouzezahrai , M. Rahimi Azghadi and D. Vazquez, "Weakly Supervised Underwater Fish Segmentation Using Anity LCFCN," *Scientific Reports,* vol. 11, no. 1, pp. 1-10, 2021.

[16] C. Robinson, A. Ortiz, H. Hughey, J. A. Stabach and J. M. Lavista Ferres, "Detecting Cattle and Elk in the Wild from Space," *arXiv preprint arXiv:2106.15448,* 2021.

[17] R. Guerrero-Gómez-Olmedo, B. Torre-Jiménez, R. López-Sastre, S. Maldonado-Bascón and D. Onoro-Rubio, "Extremely overlapping vehicle counting," in *In Iberian Conference on Pattern Recognition and Image Analysis*, 2015.