Image Segmentation and Distribution Metrics For Chicken Welfare Monitoring

Georgia Tech OMSA Practicum Spring 2022

Joseph John Janicki, Jun Xiong Tan, Liuhui Zhao @ AudioT - V2

This study examined video analytics methods applied to videos of chickens with the ultimate goal of automated monitoring of animal welfare. Based on the premise that birds will typically space themselves out and extend their wings when environmental conditions are hot/humid, and cluster together when cold, we proposed several metrics to track the spatial distribution of chickens over time to assist early mitigation efforts if any unfavorable environmental conditions or abnormal behaviors are detected. The process consists of two parts: segmentation of chickens from other objects in video frames, and deriving distribution metrics from the segmented video frames. Three different segmentation techniques - local thresholding, Mask R-CNN, and YOLOv5, were explored and compared. The results suggested that, dependent upon the distribution metrics used to indicate chicken behavior, each segmentation technique has its unique advantages and shortcomings. We demonstrated that different distribution metrics have their own merits in reflecting different aspects of chicken behavior: 1) total occupancy indicates the number of chicken during a certain time period, which could also be adapted to a specific area of interest such as feeder utilization; 2) zonal occupancy and cluster analysis can be used to determine the spatial variation of chicken clusters, where abnormal clustering behavior over a certain area might be an indicator of potential problems; 3) occupancy heatmap provides a direct visualization of the changes in occupancy over time and space, such as hot-spots where chickens tend to stay over a long time period, and chicken motion paths over a short time period.

1 INTRODUCTION

Video analytics has been widely applied in different fields, yet the research activity in animal welfare is relatively low compared to other fields. Our practicum sponsor, AudioT, focuses on 'developing methods of automatically learning audio environments and detecting anomalies including instruction, equipment failures, and changes in chicken behavior' [1]. The video data captured in the past have been underutilized. In the past cohort, major efforts were devoted to chicken detection with different techniques such as KMeans clustering, semantic segmentation, and object detection. Building upon the experience and findings from the previous cohort, we extend the detection work and also move further towards chicken welfare metric development. The main goal of our study is to explore potential metrics that could be developed from video analytics and applied towards chicken welfare evaluation. We first present a brief summary of findings from the review of relevant studies.

Ju et al. proposed an object detector and tracker for monitoring turkey welfare [3]. The detector, which is based on YOLO (You Only Look Once) v3, is applied to determine the total number of turkeys and identify individual turkeys. The tracker, based on CSRDCF, assigns separate tracking for each turkey and updates per frame. The performance of the system is tested against precision, success, and size-consistency. Similarly, a most recent study by Neethirajan applied YOLOv5 and Kalman filter for chicken detection and tracking [5]. Chicken migration patterns can be evaluated with the trajectory results from tracking individual chickens.

Anomalous chicken behavior was analyzed using computer vision in the study conducted by Kashiha et al., where recordings from top-view cameras were processed to detect problems (e.g., feeder malfunction) in terms of chicken distribution index [4]. An image pre-processing system was equipped in a commercial broiler house, and an animal distribution index was calculated to monitor chicken welfare. To calculate the distribution index, the captured image was divided to 60 zones and the occupation density per zone was computed. The distribution index was the percentage of zones whose occupation density largely deviated from the average level. A real-time linear-model based monitoring algorithm was also developed to capture certain events in the house (e.g., light change, feed type change, human interaction, etc.).

In this project, we have focused on detecting individual chickens from video clips and subsequently deriving spatial distribution metrics that can be used to evaluate chicken welfare. Thus, two main efforts are made in this study: 1) chicken detection, for which three different approaches are applied and compared (local thresholding, instance segmentation, and object bounding box detection); 2) spatial distribution metrics, for which we explore different metrics and select four indicators for chicken behavior monitoring, including total occupancy, zonal occupancy, chicken clustering, and heatmapping.

The report is organized as follows. In Section 2, we first detail the three chicken detection approaches applied in this study, discussing the procedures and data requirements. Then, we present the spatial distribution metrics with the rationale of metric selection. In Section 3, we present the results from the various chicken detection approaches and compare the performance on a different test image. After that, the distribution results are discussed in details with a set of sample videos. Finally, in Section 4, the findings from this project and suggestions for future work are included.

2 METHODOLOGY

In order to calculate any metric, there is a need to first perform chicken detection and isolate individual birds as best as possible. The pixels in an image that represent chickens should be differentiated from pixels related to the background or other objects. Broadly, our methodology consists of four parts:

I. Extracting images from video files
II. Applying detection techniques to the images
III. Calculating metrics, and
IV. Creating visualizations and reporting for the tracking of the metrics over the course of the video

2.1 Chicken detection

In this project, the chicken detection results will be the basis for computing any metrics. Therefore downstream spatial distribution analysis is heavily dependent on the accuracy of bird detection techniques. We experimented with different methods against a set of metrics and evaluated their downstream operation performance as described in the following sections.

2.1.1 Local/Adaptive Threshold Segmentation. The purpose of segmentation is to separate foreground objects from background objects, and/or to distinguish between different objects in an image. One method that was attempted for segmenting chickens from the background of the image was local threshold segmentation. A threshold is useful in the creation of image segmentation masks. For example, pixel intensities above a calculated threshold may be assigned a value of 1 and those below the threshold a value of 0. Local threshold algorithms analyze the local area around an image pixel when calculating a threshold value. This helps to appropriately segment images with varying lighting conditions. Multiple local threshold algorithms included in the scikit-image library [7] from the skimage.filters module were tested in this study.

An optimization algorithm was created to find the most ideal filter to use as well as what the ideal parameters are both for the filters and parts of the image processing pipeline.

Given the segmentation of a single image, the image processing pipeline is as follows:

(1) A raw image is read in as grayscale (0-255 intensities)

(2) A "ground truth" binary mask of that image is also read. This image was created manually via the GNU Image Manipulation Program (GIMP) by coloring in chicken pixels on a transparent layer and then saving only that layer.
(3) Histogram equalization is applied to the image.
(4) A Gaussian filter with an optimization parameter σ is applied to the image.
(5) The following filters (with various unique parameters to optimize) are applied to the image to obtain a threshold array:
    (a) threshold_local, Parameters = block_size, method, offset
    (b) threshold_niblack, Parameters = window_size, k
    (c) threshold_sauvola, Parameters = window_size, k, r
(6) A mask is created by assigning a value of 1 where a given pixel is greater than its corresponding threshold and 0 otherwise.
(7) The image is labeled. The labeling process consists of finding each island of connected pixels. For each island, a unique integer is assigned from 0 to N Islands - 1.
(8) Objects that are below a certain size, in number of pixels, are eliminated from the image. This helps to remove false positive segments that are part of the image background.
(9) Holes (islands of 0 values contained completely within a labeled island) present in the image are filled in with the label value.
(10) The label image is transformed back to a binary image.
(11) The binary image resulting from the above process is compared to the ground truth image. The comparison algorithm is as follows:
    (a) Pixels with a value of 1 in the processed image that are 0 in the ground truth image are counted.
    (b) Pixels with a value of 1 in the ground truth image that are 0 in the processed image are counted.
    (c) The sum of the preceding sums is calculated. If a processed image perfectly matches the ground truth image, the value for this metric would be 0. Values greater than 0 represent increasing error in the segmentation mask. This represents the number of false positive + false negative values.
(12) The comparison metric and tested parameter configurations are collected over a parameter and filter grid search. The optimal parameter and filter combination(s) is that with the lowest comparison metric.

Optimal parameters can change for different cameras and/or lighting conditions, and therefore it is recommended that the optimization is run for new images. Once an optimal algorithm and parameter set are obtained, methods to apply those functions to a video were created to create segmentation frames. An example of the performance of this methodology can be found in the Results (3) section.

Optionally, a foreground mask, consisting of a binary image representing feeders, waterlines, and other equipment can be provided to the code. These images can then be used to subtract out foreground objects that are not chickens. This is possible since the cameras and non-chicken foreground objects are relatively stationary. These masks are smoothed with a Gaussian filter (sigma = 0.6 here, but may be tuned depending on resolution). Figure 1 demonstrates this process.

2.1.2 Instance Segmentation with Mask R-CNN. The deep learning method was also explored to identify chickens from a set of images. Since individual chicken identification might be of interest from the standpoint of some spatial distribution metrics, the Mask R-CNN model is our choice for instance segmentation. ResNet50 was used as the backbone network for the Mask R-CNN model

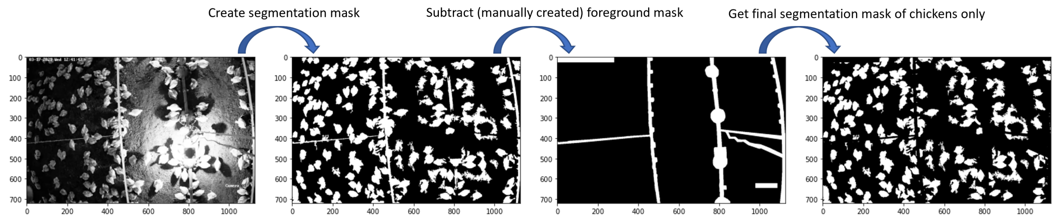


Fig. 1. Optional process of subtracting non-chicken foreground objects.

with fine-tuning on a hand-labeled data set. Different sets of hyper-parameters were also explored in this process. The model training results are presented in Section 3.

A set of cropped images from a sample video collected from a research farm was used to create ground truth labels, including 40 images for training, 15 images for validation, and 5 images for testing. The ground truth labels were hand-drawn with LabelMe [8], which is able to generate bounding boxes as well as segmentation masks. Each cropped image is of size (500, 500), covering 5 to 20 chickens, and with different light conditions. Sample labeling results are shown in Figure 2.

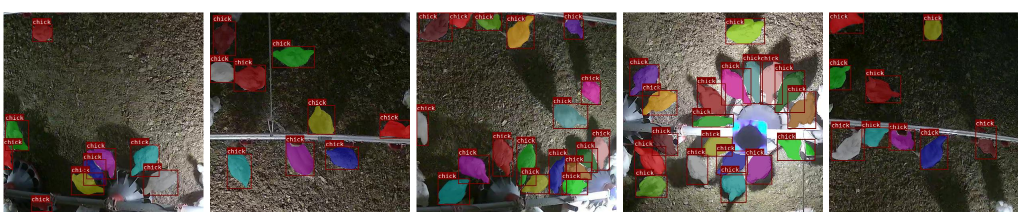


Fig. 2. Labeling results in the test dataset.

2.1.3 Object Detection with YOLOv5. The rationale for performing bird detection using an object detection model was to try and tackle situations whereby the chickens are overlapping with each other, or are very close to each other with no background pixels in between them. Segmentation masks potentially struggle with differentiating pixels from different chickens within a single blob. Bounding boxes for individual chickens might help to teach the model where one chicken ends and another begins, as well as the appropriate shapes for a chicken.

(1) Data

Ground truth labels were obtained by hand-labeling 11 full-sized image frames (2592 x 1944), with each image containing about 150 chickens, for a total of roughly 1650 bounding box labels. A test image from a separate video of size 1920 x 1080 was also labeled for unbiased test performance evaluation. Roboflow[1] was used for data augmentation. In order to train a more robust model, different augmentations were combined within the same image. However under the free plan one image can only output 3 augmented images at a time. As a workaround, different augemented datasets were generated using combinations of augmentation techniques. The augmentations used are as follows:

I. Bounding box-level noise (up to 3% of pixels)
II. Bounding box-level blur (up to 10.25 pixels)
III. Bounding box-level exposure (between -20% and +20%)

[1]https://roboflow.com/

IV. Bounding box-level brightness (between -25% and +25%)
V. Image cutout (20 boxes with 2% size each)

The combinations of augmentations used to generate augmented images are:

. I + II + III
. I + II + IV
. I + III + IV
. I + III + V
. I + IV + V
. II + III + IV
. II + III + V
. II + IV + V
. III + IV + V

where each original image generated 3 augmented images for each combination of augmentations. The resulting training dataset (generated from 9 original images) consists of 252 images resized to (1952 x 1952). The validation dataset consists of 2 original images as well as 2 augmented images. The validation dataset includes augmented images as we want to choose a model that is also able to generalize and perform well in less than ideal situations. Figure 3 shows a sample with augmentations I, II and V present.

Fig. 3. Partial sample of image with noise, blur and cutout

(2) *Model*
The YOLO family of object detection architectures is popular due to their inference speed and good performance, making them suitable for deployment for real-time inference on smaller devices. The version used here is YOLOv5, introduced by Ultralytics in an easy-to-use open-source repository [6] using the Pytorch framework. Models of different sizes pre-trained on the COCO 2017 dataset [2] are available. Due to GPU memory constraints, only the YOLOv5s (7.2M parameters) and YOLOv5m (21.2M parameters) models were experimented with.

(3) *Hardware*
Model training was conducted using an AWS EC2 g4dn.4xlarge instance which comes with 16 GiB of GPU memory.

2.2 Distribution Metrics

Based on the literature review, we propose a set of distribution metrics to measure chicken distribution in the monitoring area, which indirectly reflects chicken welfare and environmental conditions.

2.2.1 Total Occupancy. Total occupancy is calculated by the ratio of total chicken pixels over the total image pixels. Further, over a series of images, the average filled area per chicken can be computed, which could be used to transform the pixel unit to chicken unit (i.e., translate total occupancy from percentage to total number of chickens in an image). Consequently, the change in total occupancy over time serves as a rough indicator of chicken moving in and out of the scene. This metric could also be easily adapted to an occupancy over a specific area of interest, such as a feeding station, or a water line segment.

2.2.2 Zonal Occupancy. Although the total occupancy reflects the general number of chickens in the scene, it does not imply any spatial pattern in the monitoring area. Alternatively, the zonal occupancy, which calculates occupancy value per zone, is useful in identifying both the spatial pattern and the temporal trend in different zones of interest in the entire monitoring area. For this purpose, the maximum number of chicken could be estimated based on average chicken size and the zonal area. Then, the zonal occupancy could be represented by the percentage of total number of chickens in a zone over the maximum number of chicken the zone could accommodate.

2.2.3 Chicken Clusters. Chicken clusters can be used to measure how chickens are grouped or separated spatially in a scene, which can be used as a proxy indicator for environmental conditions. Clumping behavior may indicate a cold condition, whereas dispersed patterns may reflect a high temperature and/or humidity condition. To get an estimation of the number and size of chicken clusters in an image, an undirected graph is built from chicken detections, with each node representing one detected chicken, and each edge representing a pairwise distance within a specified threshold distance. As the pixel unit can be translated into chicken unit, the distance can also be represented by chicken distance (in terms of average chicken width or chicken length). Network statistics such as node degree and average clustering coefficient can be derived. Finally, the number of chicken clusters is estimated through the number of connected chicken/chicken groups.

2.2.4 Occupancy Heatmap. Occupancy heatmaps were produced by taking a series of segmentation result frames over some time period and sampling frequency, adding up the values in all segmentation masks, scaling resulting sums to 0-255 (8 bit image), and displayed using perceptually uniform sequential colormaps, such as viridis. Available color maps are described here. When multiple subsets of frames from different time points in a video are created, differences in the heatmaps can be taken by subtracting the values in one heatmap from another. The heatmap differences can be used to aid in detecting changes in occupancy and activity over time.

3 RESULTS

In this section, both the segmentation and distribution results are presented. We discuss the segmentation performance for each detection method outlined in Section 2 and compare the three approaches against one test image.

3.1 Segmentation Results

A set of performance metrics is used to evaluate the segmentation results. Based on the test image, we listed the performance of the above-mentioned segmentation methods. To calculate the performance metrics, we have the following basic measurements defined.

(1) N: total number of pixels in an image.
(2) N_{tp}: number of true positives, i.e., total number of pixels with correctly detected chickens.
(3) N_{fp}: number of false positives, i.e., total number of non-chicken pixels that are detected as chickens.
(4) N_{tn}: number of true negatives, i.e., total number of correctly detected non-chicken pixels.
(5) N_{fn}: number of false negatives, i.e., total number of chicken pixels that are detected as non-chickens.

Table 1. Performance metrics for evaluating segmentation

Metric	Definition	Description
Accuracy	$\frac{N_{tp}+N_{tn}}{N}$	Fraction of correctly detected pixels
Precision	$\frac{N_{tp}}{N_{tp}+N_{fp}}$	Correctly detected chicken pixels over detected chicken pixels
Recall	$\frac{N_{tp}}{N_{tp}+N_{fn}}$	Correctly detected chicken pixels over true chicken pixels
Dice	$\frac{2N_{tp}}{2N_{tp}+N_{fp}+N_{fn}}$	Two times the intersection divided by the sum of chicken pixels
IOU	$\frac{N_{tp}}{N_{tp}+N_{fp}+N_{fn}}$	Intersection over Union

3.1.1 Optimized Local Thresholding. Optimal parameters were identified for each of the Local, Sauvola, and Niblack thresholding methods. Figure 4 shows an example of optimization results as an initial evaluation using 24,000 parameter combinations. Ultimately, larger parameter grids were tested and results are described in Table 2.

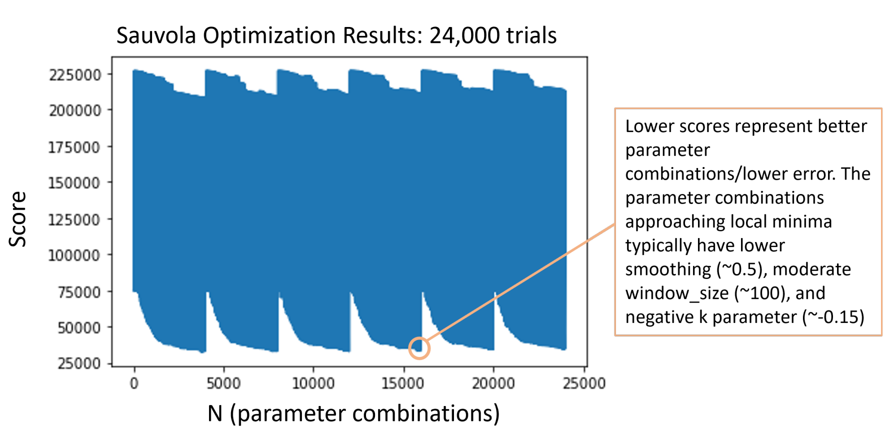


Fig. 4. Sauvola local thresholding optimization results.

Table 2. Local Thresholding Optimization Results

Method	N Combos Tested	Avg Time(s)/Combo	Best Score	Best Parameters
Local	1,440	0.61	165,167	'block_size': 49, 'method': 'mean', 'offset': 0.1, 'sigma': 0.75
Niblack	50,000	0.15	61,241	'window_size': 179, 'k': -0.7, 'sigma': 0.25
Sauvola	75,000	0.12	55,485	'window_size': 93, 'k': -0.15, 'r': 1.0, 'sigma': 0.5

The Sauvola algorithm completes in the fastest time and has the best score of the three algorithms tested. The Niblack algorithm had similar performance, and the Local algorithm performed worst. The original, ground truth, and optimal images for each algorithm are shown in Figure 5. Note that the ground truth for the optimization includes the full foreground; this generally performs better during the optimization phase, and later analysis and processing using those optimal paramaters can take advantage of the background subtraction methodology per the Methods (2) section.

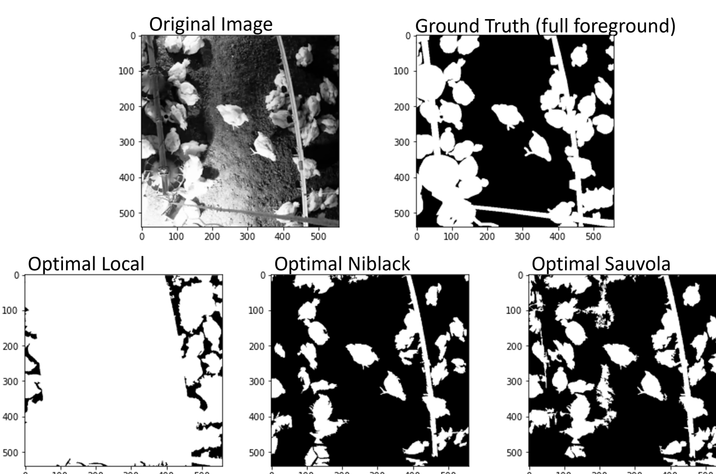


Fig. 5. Local thresholding optimization results.

The optimal Sauvola parameters were applied to frames of a video (with a different resolution) that was not used for obtaining the optimal parameters, and the results are shown in Figure 6. While this method is able to identify most chickens, it may miss some around particularly bright areas. Table 3 shows the resulting metrics for this example. While performance is generally not as strong as e.g. Mask R-CNN, this methodology requires less manual work for image labeling/training.

Table 3. Model performance on the test image with local (Sauvola) thresholding

Accuracy	Precision	Recall	Dice	IOU
0.90	0.85	0.73	0.78	0.64

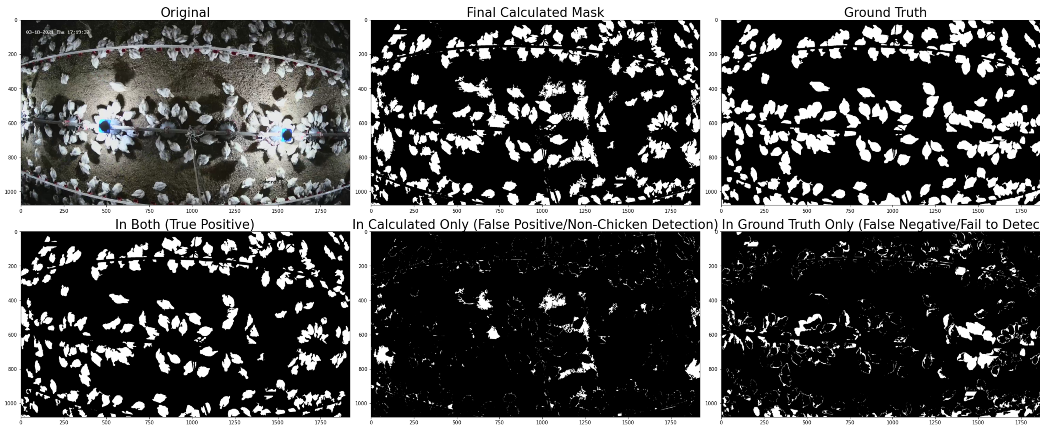


Fig. 6. Performance of Sauvola applied to video not used for obtaining parameters

3.1.2 Instance Segmentation with Mask R-CNN. Different optimizers, learning rates, and training epochs were tested in the model training process. The final model uses SGD optimizer with initial learning rate of 0.005, and a stepLR learning rate scheduler which decays the learning rate by 0.1 every 3 epochs. A total of 20 epochs was trained with a set of random transformation of the input images, including horizontal flipping, photo-metric distortion, and zooming out.

Overall, the model is able to achieve an accuracy of 0.98, a precision of 0.93, a dice of 0.94 and an IOU of 0.89 on the test image set from the same video source. It should be noted that the test images come from the same video as the training and validation images, thus we see a high detection accuracy in the final results. Applying the model to a larger image from the same video, it is observed that most chickens in the image were detected correctly, although there are a few misses in some areas, as shown in Figure 7.

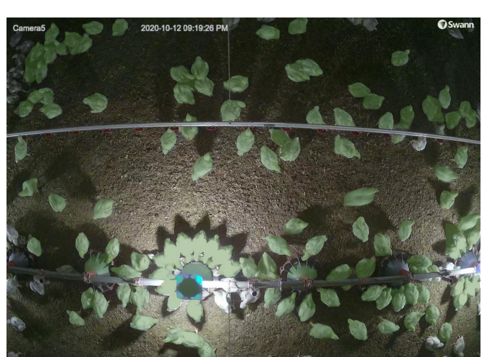


Fig. 7. Detection results with a larger image.

With the test image as shown in Figure 6 (i.e., an image from another video source), the detection results (with true positives, false positives, and false negatives) with Mask R-CNN are shown in Figure 8, and the metrics in Table 4. Major false positives come from classifying shadows as chicken pixels, and major false negatives are located along water lines, and in corner areas where heavy distortion is observed. Overall, the model achieves an accuracy of 0.88, and a dice of 0.73 on the image from a different video source.

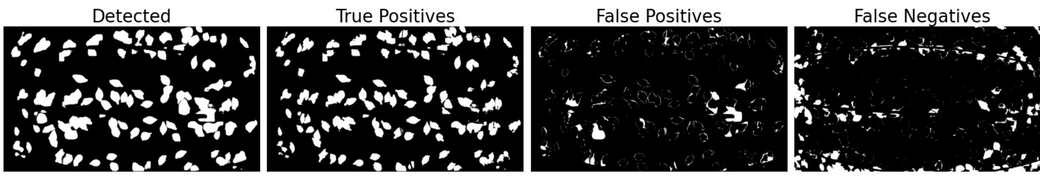


Fig. 8. Detection results from Mask R-CNN.

Table 4. Model performance on the test image with Mask R-CNN

Accuracy	Precision	Recall	Dice	IOU
0.88	0.85	0.65	0.73	0.58

3.1.3 Object Detection with YOLO v5. In computer vision, mean Average Precision (mAP) is a common metric for evaluating the performance of object detection models that takes into account both localization and classification capabilities. AP computes the average precision value for recall value over 0 to 1. It can also be calculated by finding the area under the precision-recall curve.

Object detection models make predictions in terms of bounding boxes and class labels. Bounding box predictions are evaluated by their overlap with ground truth bounding boxes, using the metric intersection over union (IoU). For each IoU threshold, an AP value can be calculated. The mAP is therefore obtained by taking the mean of AP values across IoU thresholds, though the definition may vary. For the COCO 2017 challenge, the mAP is averaged over all object categories and 10 IoU thresholds. For example, $\mathrm{mAP}^{\mathrm{val}}$@[0.5:0.95] corresponds to the mean AP for IoU from 0.5 to 0.95 with a step size of 0.05. The performance of the YOLOv5 models used in this project on the COCO val2017 dataset is shown in Table 5.

Table 5. COCO val2017 mAP of YOLOv5 models [6]

Model	parameters (M)	$\mathrm{mAP}^{\mathrm{val}}$@[0.5]	$\mathrm{mAP}^{\mathrm{val}}$@[0.5:0.95]
YOLOv5s	7.2	0.568	0.374
YOLOv5m	21.2	0.641	0.454

Both models were trained to 200 epochs on our custom dataset with the default hyperparameter settings[2]. The validation results during training are shown in Figure 9.

The YOLOv5s model was able to achieve **mAP@[0.5] of 0.975** and **mAP@[0.5:0.95] of 0.712**, while the YOLOv5m model was able to achieve **mAP@[0.5] of 0.976** and **mAP@[0.5:0.95] of 0.717**. Based on the training/validation results, the smaller YOLOv5s model will be used in this study as it is almost as performant as the YOLOv5m model and the difference is not significant. A larger model has more parameters, requires more memory, and is slower to run. Figure 10 shows the predictions of the YOLOv5s model on a validation image with a confidence threshold of 0.5.

[2]https://github.com/ultralytics/yolov5/blob/master/data/hyps/hyp.scratch-high.yaml

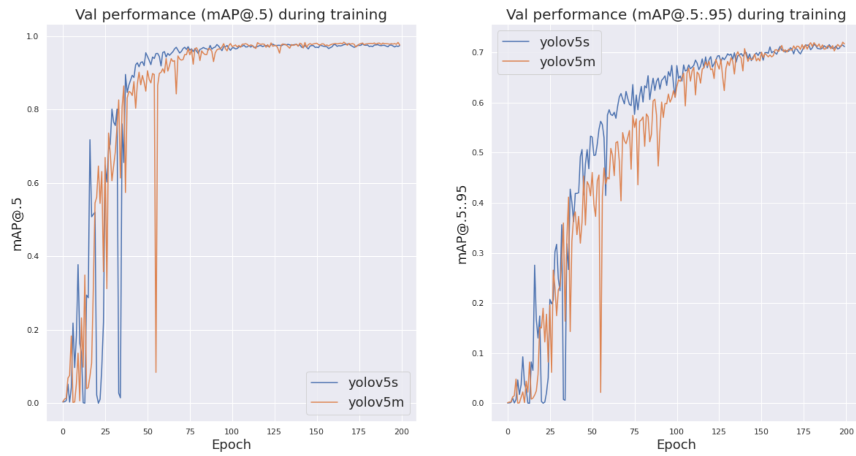


Fig. 9. Validation results of YOLOv5 model training

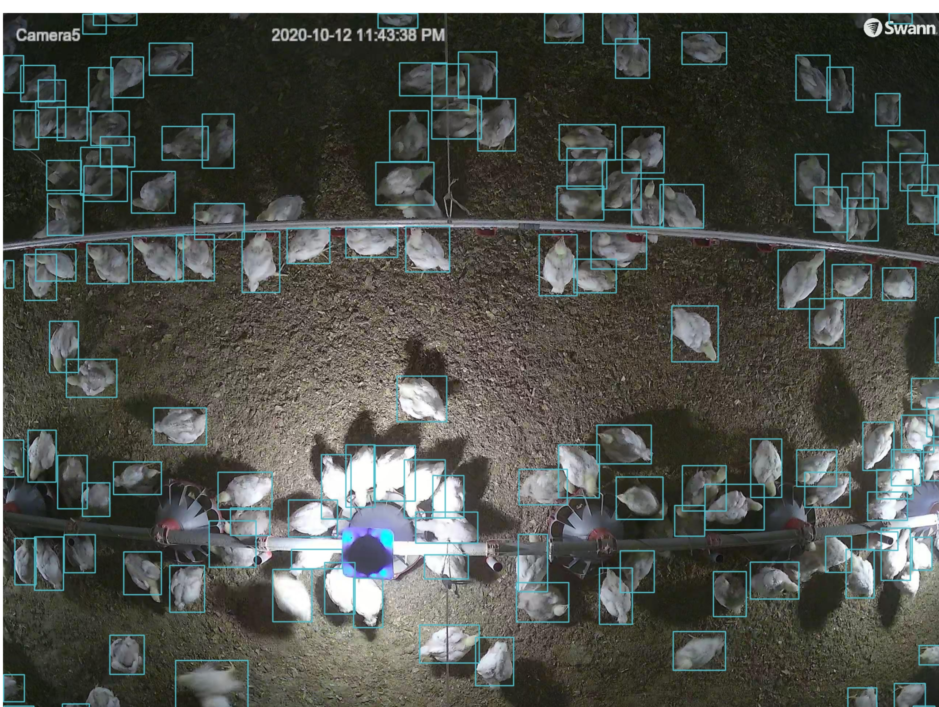


Fig. 10. YOLOv5s validation predictions

The test performance of the model was also evaluated using an image from another video source. Figure 11 shows the test image bounding box predictions using the same confidence threshold. The performance metrics are quantified in Table 6.

Fig. 11. YOLOv5s test predictions

Table 6. YOLOv5s test image results

True positive	False positive	True negative	False negative	Precision	Recall
211	3	0	38	0.986	0847

The model shows a lesser ability to discern adjacent chickens from each other when the lighting is not optimal and the image is blurry, especially at the image fringes, though that is also something a human would struggle with. The model is also able to localize chickens under conditions of extreme brightness near lamps where the chicken pixels are almost entirely white, indicating an ability to use shape/chicken outlines for identification. Figure 12 shows samples of good predictions under more challenging conditions.

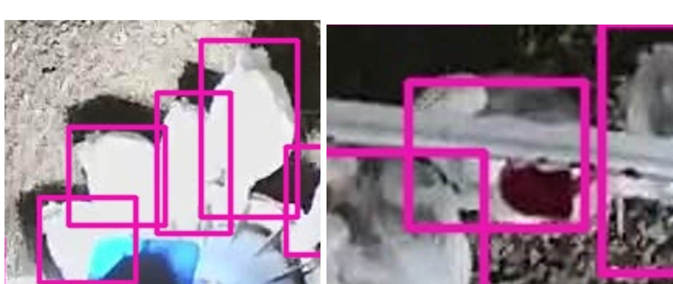


Fig. 12. Samples of test image predictions under bright conditions (left) and occlusion (right)

3.2 Distribution Results

In this section, we present the results and analysis of the distribution metrics as defined in Section 2.

3.2.1 Analysis on total occupancy.

(1) **Changes within an hour:** Considering that the detection results from Mask R-CNN may inaccurately group several chickens into one segmentation, the number of chickens in each identified area needs to be estimated in order to count the number of chickens. With a test video capturing 1 hour of chicken activity between 9:20PM and 10:20PM, an image is grabbed every 10 seconds, resulting in $\sim$ 360 frames (a sample image with detection results is shown in Figure 7). An estimation of the average chicken size is calculated with the median filled area. The original image size is 2952 x 1444; the estimated median area is 9,643 pixels, with an average minor axis length of 89 pixels and an average major axis length of 150 pixels. To account for different shapes of chickens due to movement, an additional 20% is taken into account when translating pixels to number of chickens.

    The test video captures 6 feeding stations and a water line in parallel. On average, there are 108 chickens in a scene with a standard deviation of 7 chickens in an hour (Figure 13). From the seasonality panel in Figure 13, we observed that there are around 3 peaks and 3 valleys in each 10-minute interval, indicating an exchange of chickens around feeding stations/water lines. Since most of the captured area in the test video consists of feeding stations and the waterline, we can roughly estimate from the seasonality figure that chickens generally spend around 2 minutes on eating and/or drinking, although a zoomed-in view of feeding stations and/or water line segment is better for an accurate analysis of feeding/drinking behavior. In the meantime, there is also a decreasing trend reflecting the level of chicken feeding and drinking activity within the hour.

(2) **Changes across different days:** Additionally, a set of videos across different days is examined to track chicken and behavior changes across multiple days. For this experiment, 18 videos from 3 days (10/12, 10/14, 10/16) are collected, each of 1-hour length with a 1-hour gap between two consecutive monitoring records. The same hourly time periods were captured for all 3 days of monitoring. The above test video is included as the first video in this data set. For each video, an image is extracted every 10 minutes for checking the general trend on a larger temporal scale, resulting in a total of 108 images.

    Chickens were growing fast; we observed that the median filled area that is used for estimating the average chicken size is changing across the three videos. For each day, the median filled area from the extracted images is calculated to estimate the average chicken size, as well as the average length and width of a chicken. As shown in Table 7, due to chicken movement and activity level, the average chicken size over a day is slightly different from that over an hour. Nevertheless, the average size of a chicken increased rapidly over time: about 10% increase in 5 days, a 12% increase width-wise, and 20% increase area-wise in pixel units. The average number of chickens in the monitoring area is similar across different days, reflecting a relatively stable feeding and drinking behavior on a daily basis.

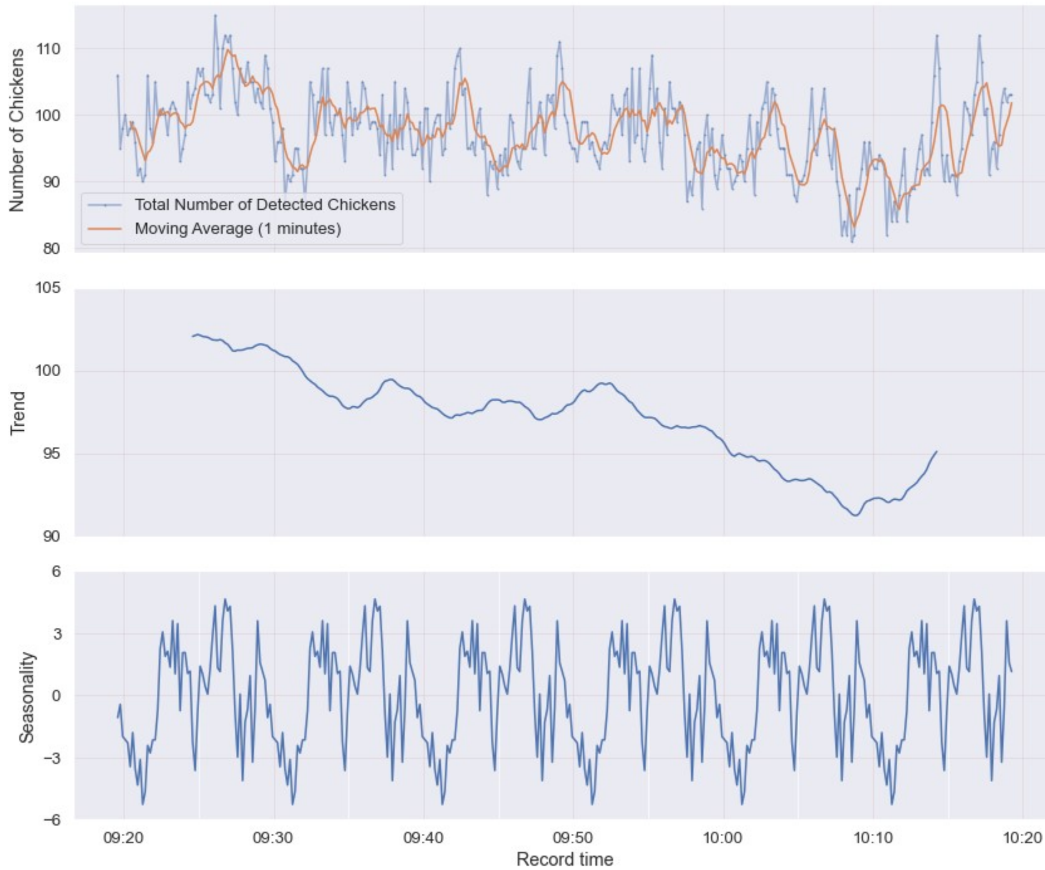


Fig. 13. Number of chickens in an hour with trend and seasonality.

Table 7. Average chicken size by pixels in different days.

Date	Size	Width	Length	Number of chickens
10/12 (first hour)	9,463	89	150	108
10/12-10/13	10,207 (base)	91	151	114
10/14-10/15	11,191 (+9.6%)	97	159	115
10/16-10/17	12,360 (+21.1%)	102	166	113

(3) **Feeder utilization:** Adapting the total occupancy to specified areas of interest, an analysis was conducted on feeder utilization using the YOLOv5s model to localize chickens. Using the coordinates of all detected chickens within an image, the chickens were counted within feeder areas marked out in Figure 14, of which there are 6 labelled F1 to F6 in the test videos used. For this section the 7am videos from 3 different days were used. Each test video spans a period of 1 hour, from which a frame is extracted every 19 seconds for a total of 193 frames per video.

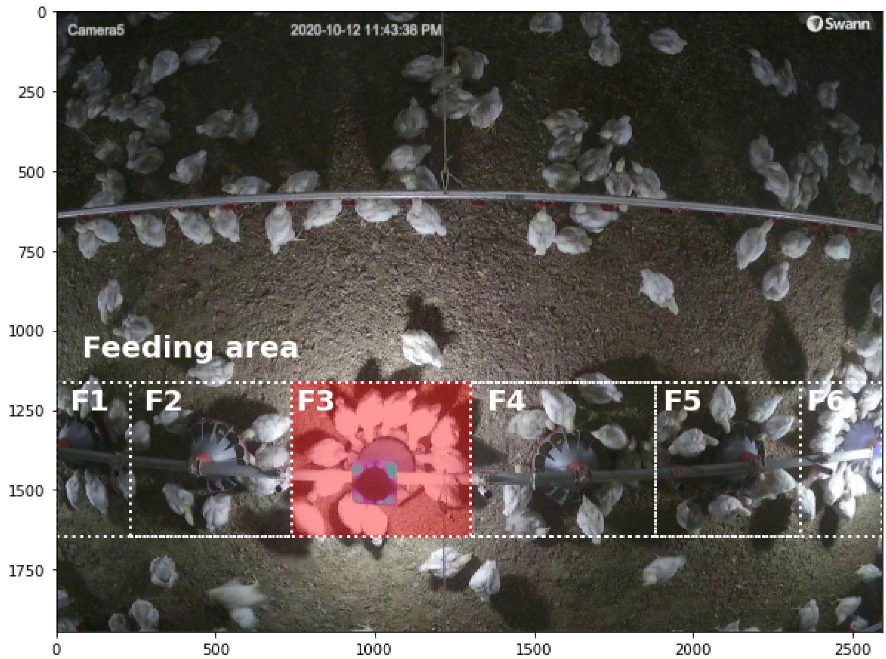


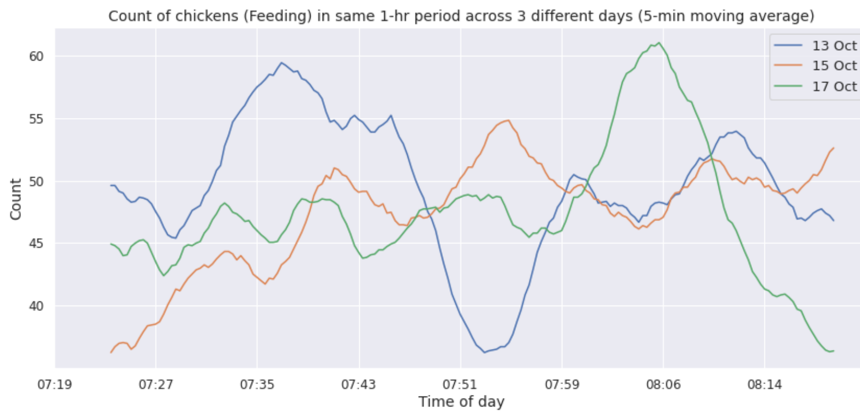
Fig. 14. Marked feeding areas

Feeder utilization across different days

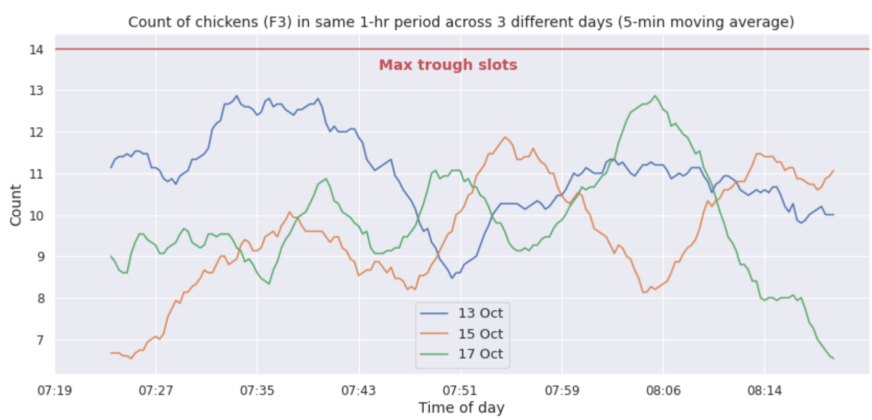
The count of chickens that were feeding in total (Figure 15a) and localized at feeder 3 (Figure 15b) within the same hour of the day was compared across 3 different days. A 5-min moving average window was used to smooth out the visualization. Although the graphs display variability both within the hour and across different days, dips/spikes in chicken count within a short time period should not be construed as conclusive indicators as they could just be normal stochastic noise. Longer term continuous data would be helpful in establishing a normal range of variability. Further context, such as whether feeding times are fixed or environmental disturbances, would be helpful additional information in evaluating feeding activity. Monitoring the activity around an individual feeder could be useful for detecting anomalies such as blockages which would likely show up as a sustained decrease in the chicken count or the sustained absence of chicken pixels. The training for the object detection model could be modified so as to focus on whether the individual feeder trough slots are occupied by chickens, thus increasing its accuracy for this specific use case.

Breakdown of individual feeder activity

Individual feeder activity can be monitored for purposes such as detecting blockages. Figure 16 shows an example of the stacked area plots of the chicken count around feeders 2 to 5 (5-minute moving average) within a 1 hour period. The plot also allows for easy monitoring of total chicken count.



(a) Total number of chickens feeding



(b) Count of chickens at F3

Fig. 15. Comparing feeder utilization across different days

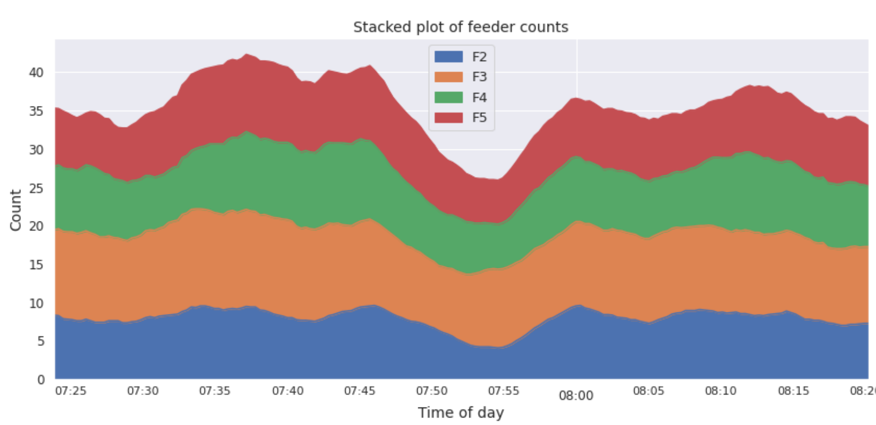


Fig. 16. Stacked area plot of feeder counts

3.2.2 Analysis on zonal occupancy. To analyze the spatial pattern in the monitoring area, the zonal occupancy was calculated for each of the 108 images used in section 3.2.1. As an example, the monitoring area was divided into 6 zones. For each zone, the maximum number of chickens the zone can handle was estimated by dividing the area of the zone with the average chicken size taking into account the relaxation ratio. Thus, the percentage zonal occupancy was prepared.

As in Figure 17, the heatmap shows both the spatial difference and temporal trend across different zones. For each recorded time point, the variations amongst the zonal occupancies are revealed along the y-axis, while for each zone, the changes in the occupancy over time is revealed along the x-axis. The average zonal occupancy reflects a general picture of chicken occupancy over time. For both individual and average zonal occupancies, the maximum values are of interest and should be noted for further analysis.

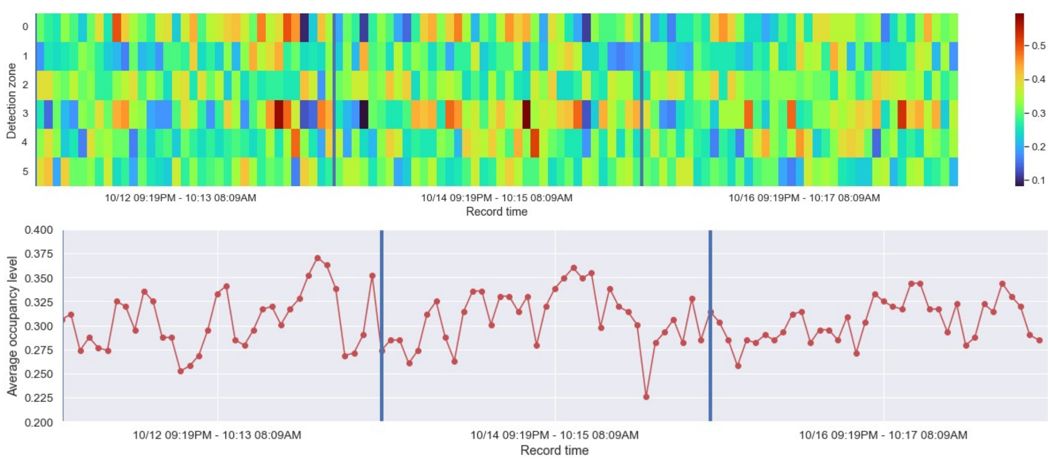


Fig. 17. Results of zonal occupancy (up: zonal occupancy vs. record time, bottom: average zonal occupancy).

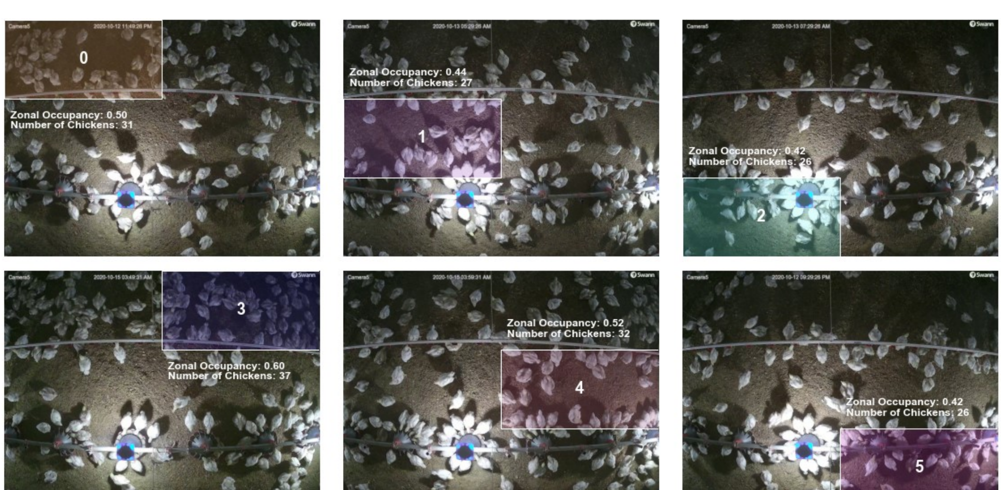


Fig. 18. Images with maximum occupancy in each zone.

For instance, Figure 18 shows the images with maximum occupancy in each zone: the maximum occupancy in Zone 0 is 0.50, containing a total of 31 chickens, whereas the maximum occupancy in Zone 3 is 0.60 with 37 chickens. Similarly, Figure 19 draws the image with the maximum average zonal occupancy (to the left) and the image with the minimum average zonal occupancy (to the right).

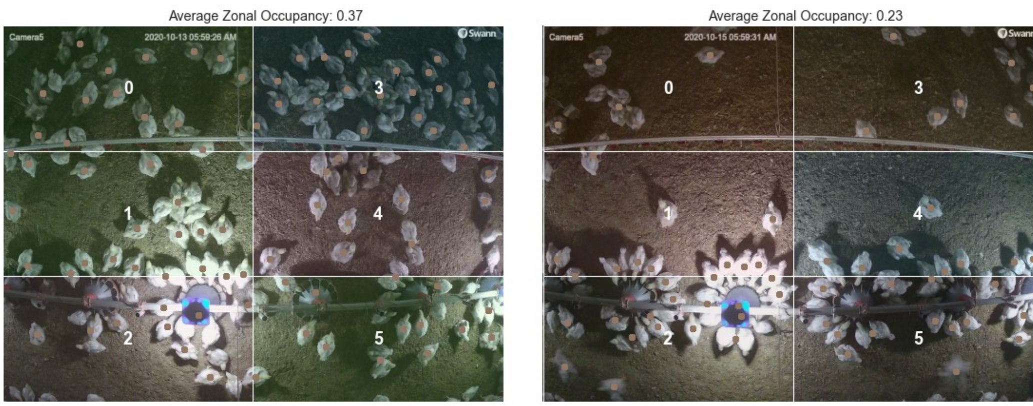


Fig. 19. Images with maximum and minimum average zonal occupancy.

3.2.3 Analysis on chicken clusters. With the estimated chicken width (i.e., median minor axis length per filled area) and chicken length (i.e., median major axis length per filled area) as a threshold for connectivity, a connected graph can be built from the Mask R-CNN segmentation results. For instance, in Figure 20, each detected chicken and/or chicken group is color coded, where each node represents the centroid of the detected area with the size relative to the number of chickens in a detected group. A link between two nodes means that these chicken (groups) are close enough to form a cluster. Considering chicken positions and size factors, the threshold for closeness was defined as the mean value of estimated chicken length and width. In total, there are 64 identified clusters in the picture, including one chicken per cluster, and the largest cluster contains 5 chickens.

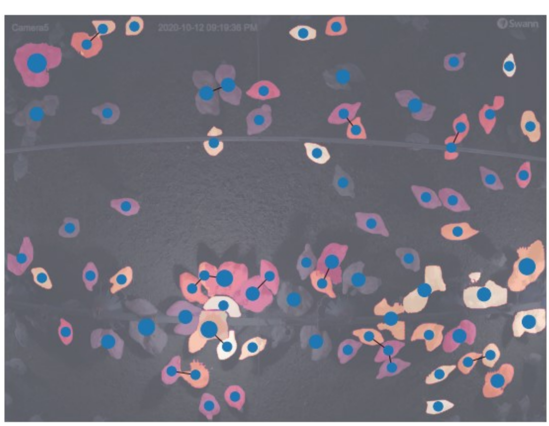


Fig. 20. Chicken clusters identified with a predefined distance threshold.

To make sense of the identified clusters, four specific zones in the monitoring area were selected for comparing the changes in the clusters, including a feeding zone, a drinking zone, a dark zone, and a light zone. Considering the shape of the feeder, the feeding zone area covering three feeders is about twice the height compared to each of the other three areas, as shown in Figure 21. For each monitoring zone, the number of chickens (whose centroids are within the zone) and thus number of clusters were computed for all the 108 images as selected in section 3.2.1.

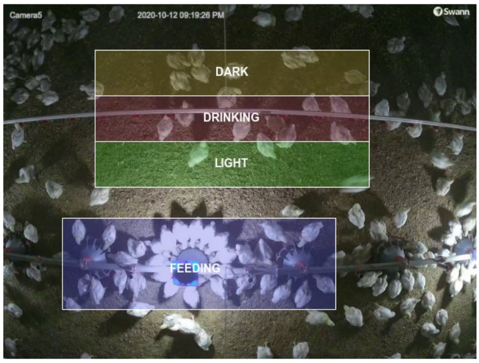


Fig. 21. Pre-defined monitoring zones.

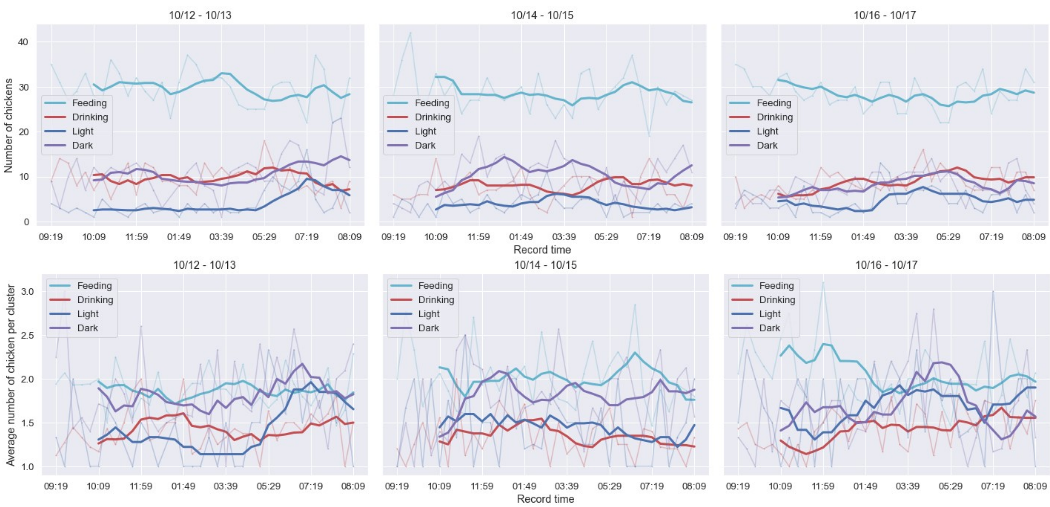


Fig. 22. Chicken and cluster trend (up: number of chickens, bottom: number of chickens per cluster).

In Figure 22, the upper panel shows the changes in number of chickens per monitoring zone over three days, and the lower panel shows the changes in the average number of chicken per cluster per monitoring zone for the same period. The thicker lines represent a 1-hour moving average from each individual data plot as shown by the thinner lines. On average, there wee about 30 chickens in the feeding zone, although there are some spikes when all three feeders were fully occupied. There were about 10 chickens in the drinking area, and it shows a clear opposite trend as compared to the feeding zone: there is generally a switch between feeding and drinking during a certain time period. Yet, it is observed that during the recording period of 12AM to 4AM on 10/15, both feeding and drinking activity declined, with more chickens resting in the dark area.

Cluster-wise, chickens were more separated from each other in the light area than in the dark area in general. Yet, on the last day of monitoring, we observed an increase in both light and dark zones, when the feeding activity seemed to drop a bit with less chickens (and clustering). Spikes in both the number of chickens and average cluster size should be given closer attention to detect early changes, such as 3:40AM to 5:00AM on 10/17. The original images with detection masks are shown in Figure 23, where different monitoring zones with high concentrations of chickens are displayed to provide a general picture of chicken density in the area during the monitoring period.

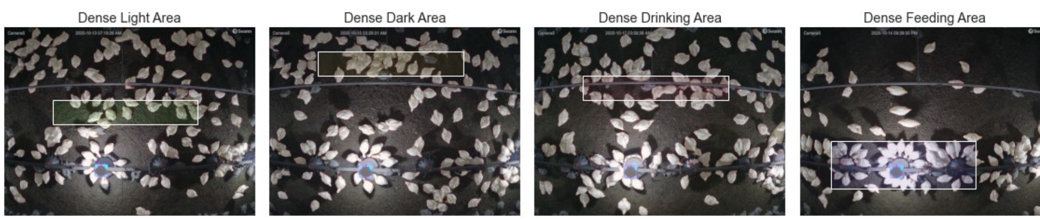


Fig. 23. Results with different chicken densities in the monitoring zones.

3.2.4 Occupancy Heatmap. Occupancy heatmaps were produced over various time periods, and the differences between those heatmaps were taken. This provides information about changes in occupancy over time. Figure 24 shows examples of the types of heatmaps that were explored.

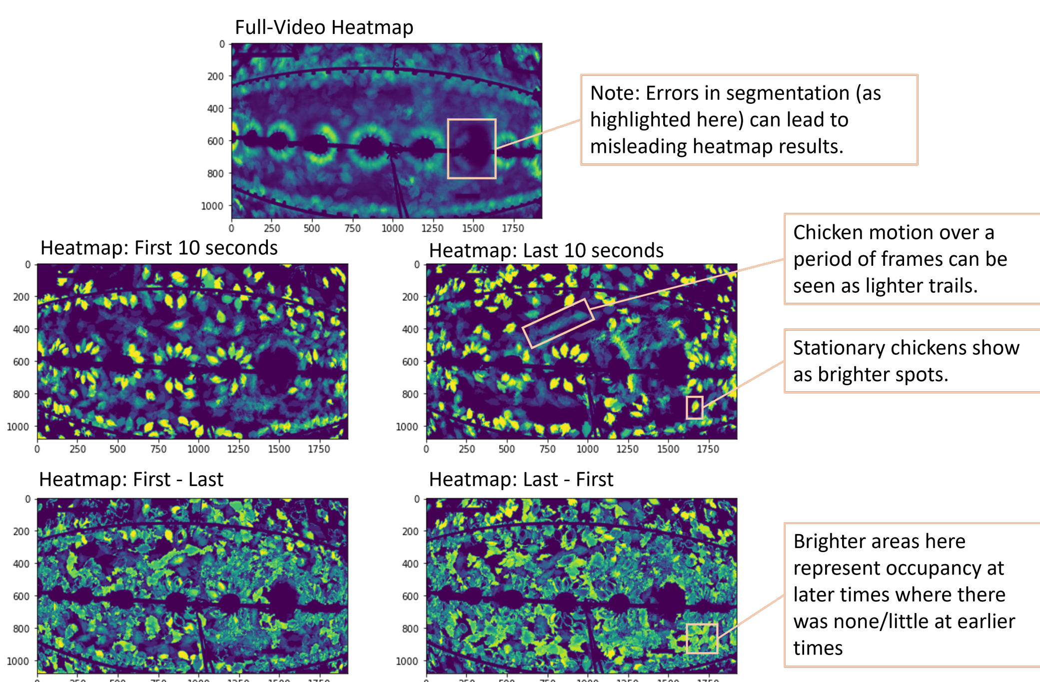


Fig. 24. Examples of Heatmaps and Heatmap Differences.

Figure 25 shows results of total occupancy applied to single frames versus heat map images. Note that these were applied on Sauvola segmentation results, in which accuracy is poor in bright areas (notably the area marked "feeding"). The heatmap images combine occupancy over some time window. In the example shown in the figure, the heatmap represents the combination of three frames, representing 30 seconds as each frame was sampled once per 10 seconds.

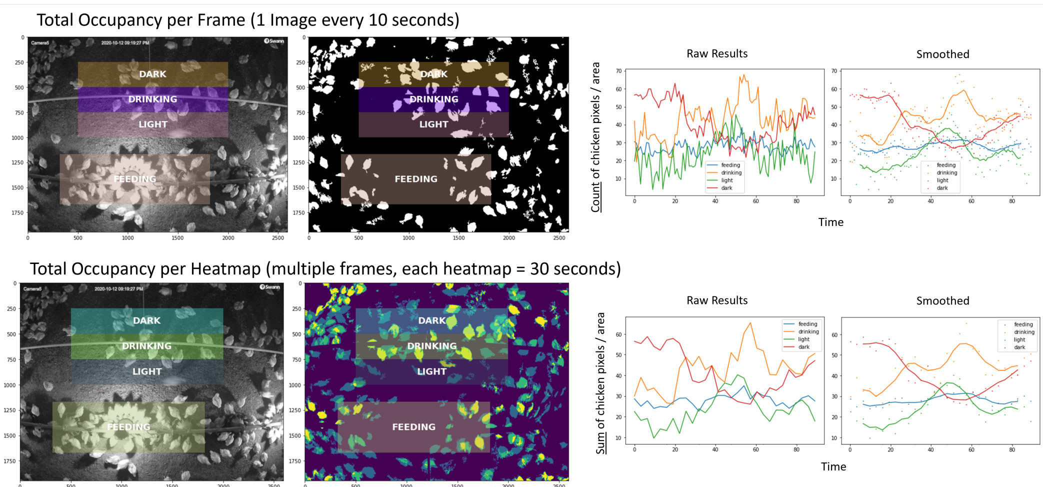


Fig. 25. Example of using heatmap images for smoothed occupancy results, which may provide utility in seeing relative trends for different areas of interest.

The heatmap images contain time-averaged data and demonstrate areas of high activity and low activity over a given period. If there are similar, high intensities across all chickens, that can indicate low levels of activity that may point to activities such as sleeping/resting, or perhaps trying to cool off if the conditions are too humid if they are spread apart or trying to warm up if they are clustered together. Conversely, if there are similar, low intensities across the image, this can indicate very high levels of activity, which can indicate that the animals are using more energy than is needed.

3.2.5 Network statistics. As a proxy for chicken environmental welfare, the spatial distribution of the chickens was analyzed using network analysis. The chickens may choose to distribute themselves differently based on environmental conditions such as temperature, humidity, lighting, sound etc. Only the top portion of the video frames was used to remove the bias from the feeding troughs. The left and right edges of the video were also cropped out for this analysis due to the image distortion which leads to inaccurate distance estimation. More ideal bigger sample population sizes can be obtained using a video feed not focused on feeding areas, as well as aggregating feeds from different cameras for a larger receptive field. The sample video used here is a 1-hour 7am video with 193 frames extracted every 19 seconds.

In order to model the chicken distribution as a graph, we first determined an arbitrary threshold distance around each identified chicken centroid (or node). The centroids were obtained from bounding boxes predicted by the YOLOv5s model. Figure 26 shows the limits of this surrounding area with a radius of 150 pixels. The unit of measurement used was pixels since the actual dimensions of objects in the video used is unknown. It is important to note factors affecting the selection of this threshold distance, such as size of chickens or any distortion/stretching in the image. This threshold distance defines the circular area around a chicken, such that if the pairwise distance with another chicken centroid falls within this threshold, the 2 chicken nodes will be connected with an edge.

Fig. 26. Threshold boundary for linking 2 chicken nodes

Using the graph constructed as above, the following descriptive statistics were computed:

- *Max node degree*
- *Average and max cluster size*: A cluster is defined as a group of at least 3 chickens. Clusters, or communities, are groups of nodes that are densely connected with each other. Clauset-Newman-Moore greedy modularity maximization was used to find the partition with the greatest modularity.
- *Average clustering coefficient*: Measure of tendency of nodes to cluster together. The statistic is the average clustering for all nodes in the graph.

Figure 27 shows a plot of the various graph statistics over a 1-hour period (7am, 13th Oct 2020). The y-axis on the left is used for the blue line plots, while the red y-axis on the right is for the average cluster coefficient. The figure shows that the various network statistics are mostly correlated, and they can be used to measure how clustered the chickens are.

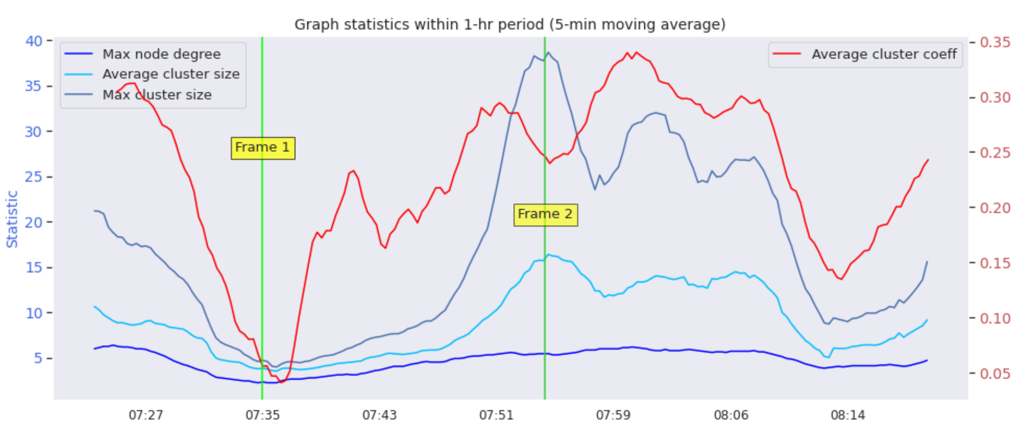


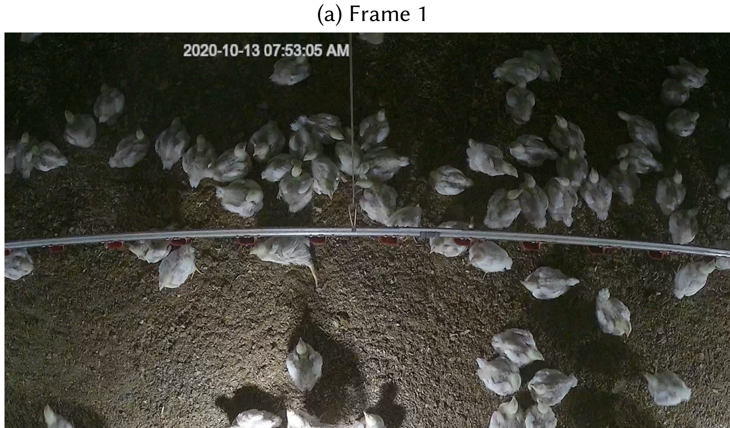
Fig. 27. Graph statistics over 1-hour period

In order to visualize how the actual distribution translates into the corresponding graph statistics, 2 image frames were extracted at the timings labeled in Figure 27. The actual frame values are shown in Table 8, and the sample frames are shown in Figure 28. The 2 images show contrasting distribution behaviour. Although frame 1 has fewer chickens, the chickens are mostly spaced apart from each other with minimal clustering, as reflected in the graph statistics. There are few connections between nodes, and detected cluster sizes are limited. On the other hand, frame 2 has lots of chickens in close proximity with each other, with a number of chickens having a high node degree and resulting in large cluster sizes.

Table 8. Graph statistics for frames 1 and 2

Frame	Max degree	Avg cluster coefficient	Avg cluster size	Max cluster size
1	2	0	3	3
2	6	0.250	14.5	49

(a) Frame 1



(b) Frame 2

Fig. 28. Cropped frames 1 & 2

4 DISCUSSIONS

In this section, we discuss the findings, limitations and future study directions based on our experience in this project.

4.1 Segmentation

(1) *Instance segmentation with Mask R-CNN:* Mask R-CNN shows its capability in automatically detecting individual objects to feed into downstream distance metric calculation. However, its accuracy depends on the model training procedure. As shown in the project, with training images from one video source, and only labeled chickens in the images, the model seems to be somehow limited to this setting without careful fine-tuning. Further data augmentation steps and hyper-parameter tuning (i.e., optimizer, learning rate scheduler, backbone, etc.) worth testing to boost detection accuracy.
(2) *Local Thresholding Segmentation:* Local thresholding segmentation fails for very bright areas of the image, but may perform well with a more-even exposure present over the image and may benefit from additional pre-processing. One of the benefits of this technique is that it does not need hand-labeled images for training aside from the foreground mask used to subtract objects that are not of interest. The method as proposed in this paper would likely not work well with non-stationary cameras.
(3) *Object detection with YOLOv5:* The model trained and used in this study was able to perform well with high precision and recall on the test images. The model was largely able to distinguish between individual chickens tightly clustered together. Inference is also fast, making it suitable for real-time deployment. However, the performance of the model was not tested in different lighting conditions (eg. nighttime) and with different chicken sizes (eg. large mature chickens), and it is likely that incorporating such images into the training dataset will result in a more robust and generalizable model. The detections can also be extended to individual object tracking via additional algorithms such as SORT (Simple Online and Real-time Tracking)[3]. Other object detection methods (single shot detector, vision transformer etc.) can also be tried to further optimize for performance.

4.2 Distribution Metrics

(1) We showed through different distribution metrics that chicken activities in the monitoring area could be detected with the segmentation results. Within a short time period, the flow of chicken around certain feeding and drinking area can be estimated, whereas over a long time period, the trends in chicken density may be tracked to provide insights in chicken behavior changes or potential environment changes.
(2) Comparing different segmentation techniques and applying them to the calculations of different spatial distribution metrics, we observed that each segmentation technique has its own advantages and shortcomings. For instance, local thresholding technique is light-weight and has no requirement for hand-labeling. It works well in generating occupancy heatmaps, but may not be an ideal option to estimate chicken numbers and track individual chicken behavior. On the other hand, deep neural network-based techniques, such as Mask R-CNN and YOLOv5, performed well in identifying and tracking individual chickens under different monitoring conditions, given enough training and fine-tuning efforts. Data augmentation was proved to be a good tool for scaling up the training dataset. Nevertheless, a large pool of diverse hand-labeled images is necessary to overcome potential overfitting problems.

[3]https://github.com/abewley/sort

(3) The threshold distance for determining whether two nodes are connected is arbitrary, and has to be adapted as the chickens change in size, or if the video settings are changed. Ideally, the sample population for graph construction should be as big as possible. This can be achieved by stitching together images from different cameras capturing adjacent patches.
(4) The network statistics computed can be utilized together with environmental readings for temperature, humidity, lighting, noise etc. to form a more robust monitoring system for chicken welfare. Sensor readings can assist in explainability with respect to how chickens are distributed. Conversely, the empirically-derived network statistics via computer vision can be used for 1) better-informed controller setpoints for environmental management systems, and 2) possibly identifying malfunctioning/inaccurate sensors.

4.3 Future Work

The results shown in this study can be extended for the purposes of specific event detection related to animal welfare, such as automated identification of behaviors such as sparring, racing, whether the animals are not eating, the duration of feeding, when and how often the animals are full/satiated, detection of non-chicken animals such as rodents that may carry disease, dust bathing, and pecking. Some additional topics for exploration would be how long the chickens are asleep/awake, activity levels, and detection of growth anomalies. Our theory is that the detection and classification of these behaviors can be most easily satisfied through reliable segmentation and classification of video frames, and through some extended inter-frame analysis which may benefit from either object tracking techniques or the use of composite heatmaps plus heatmap classification.

5 STATEMENT OF WORK

All team members contributed to the development of project concept, the review of methodology and related materials, the preliminary study of video analytics, and the final report. Each member was responsible for a different segmentation experiment and a subset of the spatial distribution metrics.

ACKNOWLEDGMENTS

The research team is thankful for all the support from the project sponsor team, including initial training, core research direction discussion, and providing necessary materials and platforms for the project.

REFERENCES

[1] AudioT. 2019. *World-class talent driving AI Innovation for Poultry Farming Applications.* http://www.audiot.ai/
[2] COCO Consortium. 2022. COCO Common Objects in Context. https://cocodataset.org/#download
[3] Shengtai Ju, Marisa A Erasmus, Amy R Reibman, and Fengqing Zhu. 2020. Video tracking to monitor turkey welfare. In *2020 IEEE Southwest Symposium on Image Analysis and Interpretation (SSIAI)*. IEEE, 50–53.
[4] Mohammadamin Kashiha, Arno Pluk, Claudia Bahr, Erik Vranken, and Daniel Berckmans. 2013. Development of an early warning system for a broiler house using computer vision. *Biosystems Engineering* 116, 1 (2013), 36–45.
[5] Suresh Neethirajan. 2022. ChickTrack-A Quantitative Tracking Tool for Measuring Chicken Activity. *Measurement* (2022), 110819.
[6] Ultralytics. 2020. YOLOv5. https://github.com/ultralytics/yolov5.
[7] Stefan Van der Walt, Johannes L Schönberger, Juan Nunez-Iglesias, François Boulogne, Joshua D Warner, Neil Yager, Emmanuelle Gouillart, and Tony Yu. 2014. scikit-image: image processing in Python. *PeerJ* 2 (2014), e453.
[8] Kentaro Wada. 2018. labelme: Image Polygonal Annotation with Python. https://github.com/wkentaro/labelme.