Video Tracking to Monitor Turkey Welfare

Shengtai Ju*, Marisa A. Erasmus†, Amy R. Reibman* and Fengqing Zhu*
*School of Electrical and Computer Engineering, †Department of Animal Sciences,
Purdue University, West Lafayette, Indiana, USA
Email: {ju10, merasmus, reibman, zhu0}@purdue.edu

***Abstract*—Although turkey production is important in the United States, few studies have focused on turkey welfare, partly because of the lack of non-invasive and automated techniques for detecting changes in turkey welfare. Disease can pose major threats to turkey welfare and human health. In this paper, we propose a novel approach for detecting and tracking turkeys in video as the first steps to monitor turkey welfare. A self-trained object detection model is used to identify turkeys in each frame of the video, and a modified object tracker is used to predict the location of each turkey in the next frame. Hand-crafted features are developed to better handle occlusion and to improve tracking accuracy. Our method demonstrates promising results when evaluated on a turkey video dataset in terms of precision, success, and size consistency.**

***Keywords*-object detection, object tracking, video analytics, turkey welfare**

I. Introduction

Turkey is a significant source of meat poultry in the United States. There has yet been little research examining turkey welfare in spite of the significance of turkey production. Disease is a main challenge for turkey production, which may cause subclinical infections that cannot be visually observed by animal caretakers. However, researchers have shown that even subclinical illness that cause inflammation and welfare concerns can be detected using bird motion in video recordings [1]. As poultry production increases worldwide, there is a need for accurate, objective, and automated monitoring of animal welfare on commercial farms to safeguard animal welfare and detect disease issues earlier.

The increasing availability of high quality, cost effective consumer grade cameras enables continuous monitoring of turkey welfare by creating a permanent record for researchers. However, given the large amount of data collected, it is not feasible for a trained analyst to manually review and annotate all video sequences. Leveraging recent advances in computer vision, we aim to develop video analytics solutions to monitor turkey behavior change and to identify subclinical illness caused by heat stress and infection. As the first steps to monitor turkey welfare from video recordings, we propose to develop methods to identify and track individual turkeys in each frame of a video in this work. Our method uses a combination of an object tracker and a detector based on the recent success of a discriminative correlation filter (CSRDCF) [2] and the YOLO detector [3]. The contribution of our work is reflected in the following aspects: (1) a novel turkey tracking system, (2) incorporating domain knowledge to improve tracker accuracy, and (3) a new evaluation metric to measure bounding box size consistency. Details of the proposed method are discussed in Section III, followed by experimental results in Section IV.

II. Related Work

A. Object Tracking

Object tracking aims to track an object or a set of objects in a sequence of frames [4]. Correlation filter trackers have shown good performances on various benchmark datasets. Examples of correlation filter trackers include Kernelized Correlation Filter Tracker (KCF) [5], Discriminative Correlation Filter Tracker with Channel and Spatial Reliability (CSRDCF) [2], and Spatially Regularized Correlation Filter Tracker (SRDCF) [6]. Correlation filter trackers use adaptive learning filters to get response maps with the estimated target patches [4]. A new target location is determined by the location of the maximum response. A correlation filter tracker is efficient because it takes advantage of the Fourier domain compared to using time domain convolution [5]. It has been shown that CSRDCF outperforms other trackers in noisy environments on the OTB2015 dataset [7]. CSRDCF introduced a spatial reliability map and channel reliability weights to constrain filter learning which enlarged the search region and improved tracking accuracy of non-rectangular objects [2].

B. Object Detection

Object detection is often used in video analytics to detect objects of interest in each frame of a video sequence [8]. It has been shown that fusing object detection with object tracking can improve the tracker's success when a long period of occlusion occurs [9]. State-of-the-art object detectors include R-CNN [10], Fast R-CNN [11], and YOLO [12]. R-CNN and Fast R-CNN extract many region proposals from an input image. Features are extracted from each region through a large convolutional neural network and used to assign a class label to each region. YOLO looks at the entire image and detects objects by dividing it into smaller regions. Region Proposal Networks, e.g., R-CNN and Fast R-CNN, require a longer time to generate detection results because an image is examined many times to generate different regions.

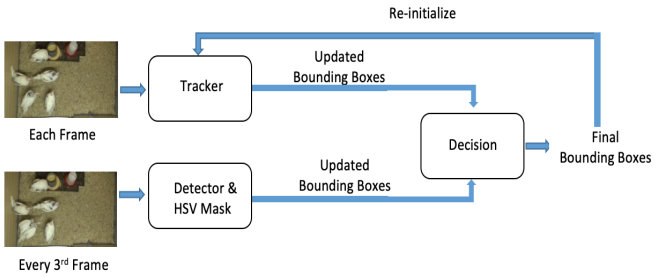


Figure 1: Overview of proposed system to track and detect individual turkey in each frame of the video.

YOLO generates detection results more quickly because it evaluates the entire image using a single neural network. In this paper, we choose to use YOLOv3 [3], third generation of YOLO, because it is efficient, accurate, and easy to implement. In the rest of this paper, we will call it YOLO for conciseness.

C. Monitoring Poultry Welfare using Computer Vision

Recently, computer vision has been applied to different tasks related with maintaining poultry wellness. The authors in [13] proposed a method for detecting a malfunctioning feeding system or drinking line in a broiler house. An automatic approach for detecting broiler lameness was proposed in [14]. Although these works contributed to improving broiler wellness, they do not target turkey welfare specifically and do not detect the health-related issues that we are interested in. Analyzing turkey behavior and interactions is crucial for understanding turkey welfare, but none of these previous works accomplished this task.

III. Methods

Our method consists of an object tracker and detector as illustrated in Fig. 1. The object tracker is based on CSRDCF [2], and the object detector is based on YOLO [3]. The detector is applied to the first frame of the video to determine the total number of turkeys (N) in the scene, and it assigns a unique ID to each turkey. We then initialize N separate trackers with the predicted bounding boxes from the detector. In subsequent frames, each tracker is updated per frame and outputs a predicted bounding box for each turkey. For every third frame, we apply the detector to update the bounding boxes.

If occlusion occurs, we find the turkeys that are isolated and assign updated bounding boxes to them based on the detector's output. For turkeys that are either partially or fully occluded, we keep their previous bounding boxes by assuming motion consistency across consecutive frames. If there is no occlusion, we assign new bounding boxes to all detected turkeys. In addition, we use the HSV color space information to determine if there is occlusion in a given frame. Since all turkeys are white in our videos, we use color to perform foreground background segmentation to separate the turkeys from the background. After obtaining bounding boxes for each turkey from the tracker, the detector, or both, we compare them to bounding boxes from the previous frame and decide whether the final bounding boxes should be updated. If the bounding box moves too much between consecutive frames, we keep the previous bounding box based on a fixed threshold. Based on the decision, we re-initialize each tracker with the updated bounding boxes.

A. CSRDCF Tracker

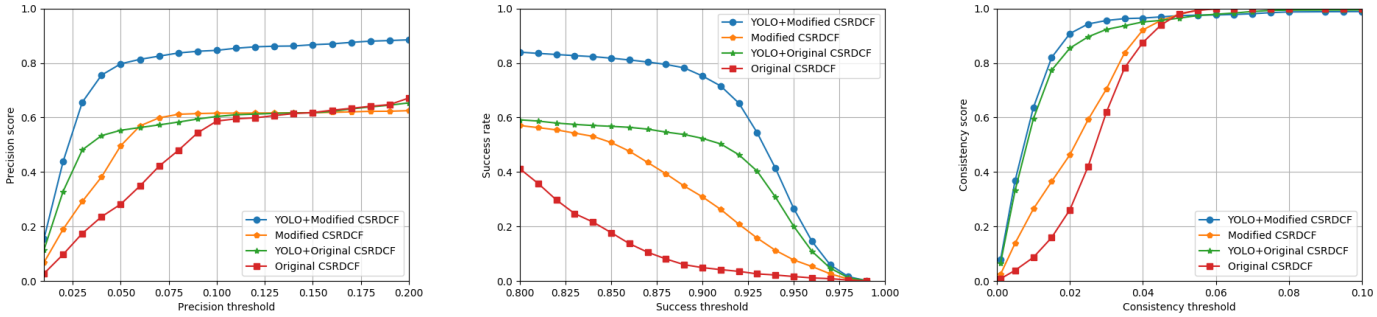
CSRDCF uses channel and spatial reliability for filter learning. Correlation responses of different feature channels are computed, and a spatial reliability map is constructed to predict pixels in the search region that likely belong to the target [2]. Channel weights are also assigned to each channel to indicate the discriminative power of an individual feature channel. Different features contribute differently to the final correlation response based on their channel reliability weights. We modified the default CSRDCF tracker parameters by lowering the filter and weights learning rate. Without lowering the learning rates, trackers often lose targets when turkeys get close to the feeding station, which indicates that the filters are too adaptive for our data. We want the trackers to focus on the turkeys instead of background objects such as the feeder and drinker. We lowered the filter learning rate from 0.02 to 0.004 and lowered the learning rate of channel reliability weights from 0.02 to 0.005. By lowering the learning rates, the trackers are more stable when occlusion occurs or when turkeys move rapidly.

B. YOLO Detector

YOLO divides the entire image into $S \times S$ grids and predicts a number of bounding boxes for each grid. Non maximum suppression (NMS) is then applied to eliminate overlapping bounding boxes with an intersection over union value greater than a set threshold. The feature extractor in YOLO consists of 53 convolutional layers called Darknet-53 [3]. Authors of [3] provide convolutional weights that are pretrained on ImageNet [15].

IV. Experimental Results

We use precision, success, and size-consistency to evaluate and compare the performance of four methods on our video data: (1) our method that combines the modified CSRDCF tracker with YOLO, (2) a modified CSRDCF tracker only, (3) a default CSRDCF tracker with YOLO, and (4) a default CSRDCF tracker only. Bounding box information is stored in the format (x, y, w, h), where (x, y) is the center of the bounding box and (w, h) is the width and height of the bounding box. The four metric values are normalized by the resolution of the video.



(a) (b) (c)

Figure 2: Evaluation Results for (a) Precison, (b) Success, and (c) Size-Consistency

A. *Dataset*

The dataset is composed of long video sequences taken by SONY Camcorders CX405 cameras at 30FPS. All videos are taken by researchers in the Department of Animal Sciences at Purdue University. Videos are taken from a small room containing five turkeys with feeding and water stations using an overhead camera placed on the ceiling. The turkeys used in our experiments are all white commercial turkeys that have been raised to 20 weeks of age. Turkeys are free to move around the room. The goal is to observe turkey behavior from these video sequences to identity potential welfare issues.

Our customized turkey detector is trained using images extracted from a 2-minute video sequence. We manually labeled bounding boxes for all turkeys in 480 frames extracted from the training video sequence. The set of frames is divided into 90% training and 10% validation. The model is trained on a single NVIDIA TITAN Xp GPU with learning rate set to 0.001. The testing video is a 3-minute video sequence recorded from the same room with the same setting as the training data.

B. *Precision*

Precision is used to measure how well our method performs in predicting the center of each turkey. We adopted a similar approach to [4]. First, we compute the Euclidean distance between the predicted bounding box and the ground truth bounding box as:

$$d_i = \sqrt{(x_p - x_g)^2 + (y_p - y_g)^2}\,, \tag{1}$$

where (x_p, y_p) corresponds to the center of predicted bounding box and (x_g, y_g) corresponds to the center of ground truth bounding box. We then introduce an indicator function Θ_i, which is assigned a value of 1 if $d_i < d_{th}$ where d_{th} is the precision threshold. The precision threshold represents the normalized distance between centers of bounding boxes. For each turkey, we sum the number of frames over N total frames, whose spatial difference with the ground truth bounding box is within the precision threshold. We then divide the sum by N and multiply by 100 to compute precision as a percentage.

Since trackers may lose targets and YOLO detector might not be able to detect all present turkeys, we keep track of precision for each turkey separately and compute the average precision score at the end. We expect a higher precision score from our method at low thresholds compared to other techniques. From Fig. 2a, we observe that combining the YOLO detector with our modified CSRDCF tracker outperforms the other methods in our dataset. By comparing the curves of the modified CSRDCF and the original CSRDCF, we see that changing the parameters greatly increases precision at lower thresholds. Also, incorporating the YOLO detector significantly increases precision as shown in Fig. 2a.

C. *Success*

During tracking, the predicted bounding boxes may capture more background information than desired even if the center of each tracked object is accurate. Therefore, a measurement is needed to tell us how well our method captures pixels corresponding to each turkey. Similar to precision, we adopted and modified the method proposed by [4]. First, we compute the intersection over union (IOU) between the predicted and the ground truth bounding boxes as:

$$\alpha_i = \frac{|b_p \bigcap b_g|}{|b_p \bigcup b_g|}\,, \tag{2}$$

where b_p is the predicted bounding box and b_g is the ground truth bounding box. Similar to precision, success for each turkey is computed as the total number of frames over N frames where α_i is greater than α_{th}. α_{th} ranges between 0 and 1, with 1 representing the perfect overlap with ground truth bounding boxes. The success score is then computed similarly to the precision score by dividing the total number of frames that satisfied the condition by N and multiplying by 100. Finally, we compute the average success score for the five turkeys in the scene.

The higher the success rate is, the better our prediction aligns with the ground truth. The goal is to achieve high

success rates at high success thresholds. From Fig. 2b, we see that YOLO combined with the modified CSRDCF tracker shows the best performance. By comparing YOLO with the modified tracker and YOLO with the default tracker, we see that object detection greatly increases the success rate. Furthermore, comparing the performance of the default tracker to the modified tracker we observe that lowering the learning rate increases the success rate at higher thresholds for our application.

D. Size-Consistency

In addition to success rate, we also include a measure of size consistency between the predicted bounding boxes and the ground truth. When trackers lose targets, the bounding boxes can become very small or large. However, the centers of those boxes might not be too far off from the actual centers of the turkeys. A predicted bounding box can have a good precision and success score, which means its center is not far from the ground truth center and the predicted bounding box has significant overlap with the ground truth. To further verify that the predicted bounding box does not contain undesired background pixels, we propose a new metric which computes the size difference between the predicted bounding box and the ground truth bounding box as:

$$c_i = |w_p * h_p - w_g * h_g|, \tag{3}$$

where (w_p, h_p) is the width and height of the prediction and (w_g, h_g) is the width and height of the ground truth bounding box. Consistency is computed in the same manner as success other than the definition of the threshold α_{th}. The threshold represents the normalized spatial average area difference of the bounding boxes.

We would like to have high consistency scores at low consistency thresholds, which means the sizes of predicted boxes are spatially consistent. From Fig. 2c, we see that YOLO combined with the modified tracker outperforms the other methods at lower thresholds, which is more desirable for our task. By comparing the curves, we see that YOLO greatly increases the consistency at lower thresholds while the modified tracker slightly increases consistency.

V. Conclusion

In this paper, we introduced a novel approach for detecting and tracking turkeys from video recordings. We fine-tuned the YOLOv3 detector by training on turkey videos from our collected data and modified the parameters of the CSRDCF tracker to adapt to our data. Our method is evaluated on a turkey video which shows promising results in terms of accuracy and consistency. However, there are several assumptions made while implementing our system. We assumed that turkeys do not move much between two consecutive frames. We also assumed that the detection results on the first frame of the video are accurate so that we can use them to initialize the trackers. Developing an automatic method for monitoring turkey welfare can greatly reduce the need for human observers, which is label intensive and error prone. Our initial results will allow us to analyze turkey behavior and interactions such as identifying aggressive or abnormal behaviors, which can indicate negative impact by environmental factors or diseases.

References

[1] S. Humphrey, G. Chaloner, K. Kemmett, N. Davidson, N. Williams, A. Kipar, T. Humphrey, and P. Wigley, "Campylobacter jejuni is not merely a commensal in commercial broiler chickens and affects bird welfare," *mBio*, vol. 5, no. 4, 2014.

[2] A. Lukežič, T. Vojir, L. Čehovin Zajc, J. Matas, and M. Kristan, "Discriminative correlation filter with channel and spatial reliability," *IEEE Conference on Computer Vision and Pattern Recognition*, 2017.

[3] J. Redmon and A. Farhadi, "Yolov3: An incremental improvement," *arXiv preprint arXiv:1804.02767*, 2018.

[4] M. Fiaz, A. Mahmood, and S. K. Jung, "Tracking noisy targets: A review of recent object tracking approaches," *arXiv preprint arXiv:1802.03098*, 2018.

[5] J. F. Henriques, R. Caseiro, P. Martins, and J. Batista, "High-speed tracking with kernelized correlation filters," *IEEE Transactions on Pattern Analysis and Machine Intelligence*, vol. 37, no. 3, pp. 583–596, 2014.

[6] M. Danelljan, G. Hager, F. Shahbaz Khan, and M. Felsberg, "Learning spatially regularized correlation filters for visual tracking," *IEEE International Conference on Computer Vision*, 2015.

[7] Y. Wu, J. Lim, and M.-H. Yang, "Object tracking benchmark," *IEEE Transactions on Pattern Analysis and Machine Intelligence*, vol. 37, no. 9, pp. 1834–1848, 2015.

[8] H. S. Parekh, D. G. Thakore, and U. K. Jaliya, "A survey on object detection and tracking methods," *International Journal of Innovative Research in Computer and Communication Engineering*, vol. 2, no. 2, pp. 2970–2979, 2014.

[9] N. Wojke, A. Bewley, and D. Paulus, "Simple online and realtime tracking with a deep association metric," *IEEE International Conference on Image Processing*, 2017.

[10] R. Girshick, J. Donahue, T. Darrell, and J. Malik, "Rich feature hierarchies for accurate object detection and semantic segmentation," *IEEE Conference on Computer Vision and Pattern Recognition*, 2014.

[11] R. Girshick, "Fast R-CNN," *IEEE International Conference on Computer Vision*, 2015.

[12] J. Redmon, S. Divvala, R. Girshick, and A. Farhadi, "You only look once: Unified, real-time object detection," *IEEE Conference on Computer Vision and Pattern Recognition*, 2016.

[13] M. Kashiha, A. Pluk, C. Bahr, E. Vranken, and D. Berckmans, "Development of an early warning system for a broiler house using computer vision," *Biosystems Engineering*, vol. 116, no. 1, pp. 36–45, 2013.

[14] A. Aydin, "Development of an early detection system for lameness of broilers using computer vision," *Computers and Electronics in Agriculture*, vol. 136, pp. 140–146, 2017.

[15] J. Deng, W. Dong, R. Socher, L.-J. Li, K. Li, and L. Fei-Fei, "ImageNet: A large-scale hierarchical image database," *IEEE Conference on Computer Vision and Pattern Recognition*, 2009.