

# Chicken Behavior Analysis for Surveillance in Poultry Farms

Abdallah Mohamed Mohialdin<sup>1</sup>, Abdullah Magdy Elbarrany<sup>2</sup>, Ayman Atia<sup>3</sup>  
HCI-LAB, Faculty of Computers and Artificial Intelligence, Helwan University<sup>3</sup>  
Faculty of Computer Science, October University for Modern Sciences and Arts (MSA)  
Giza, Egypt<sup>1,2,3</sup>

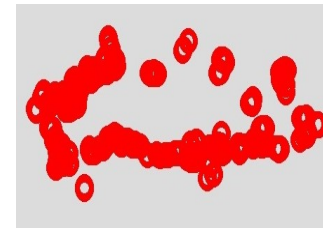
**Abstract**—Poultry farming is an important industry that provides food for a growing population. However, the welfare of the birds is a major concern, as poor living conditions leads to abnormal behavior that affects the health and productivity of the flock. In order to monitor and improve the welfare of the birds, it is important to have a surveillance system in place that monitors the behavior of the chickens and alert farmers to potential issues. This paper reviews the current state of the art in behavior analysis for surveillance in poultry farms and discuss potential future directions for research in this area. This paper presents a computer-vision-based system that detects and monitors the behaviors of the chickens in poultry farms. The system classifies three behaviors which are eating, walking and sleeping. The system takes videos as input and then classifies the behavior of the chicken. The proposed system produces an accuracy of 94.7% using Light Gradient Boosting Machine on a collected data-set of chickens, and a 98.4% accuracy on a benchmarked Human Activity Recognition data-set.

**Keywords**—Chicken; poultry; abnormal; behavior; birds

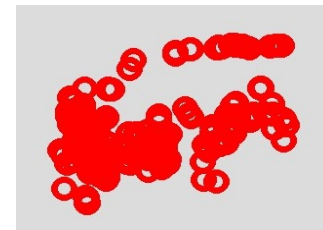
## I. INTRODUCTION

The study of chicken behavior plays a critical role in the poultry industry, as it provides valuable insights into the health and welfare of the birds. Farmers can detect any abnormal patterns that indicate illness, stress, or discomfort by analyzing chicken behaviors. This information is then used to improve their health condition, and ultimately, enhance production efficiency and profitability. In addition, an effective system for detecting and analyzing chicken behavior contributes in early disease detection, preventing outbreaks [1], ensuring food safety, protecting public health [2], and ensuring the quality of poultry production [3]. Hence, having a reliable system for monitoring chicken behavior is essential for sustainable poultry farming and ensuring the well-being of the birds.

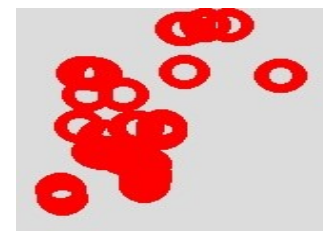
The Food Standards Agency (FSA) reported that 1.35 million chickens die prior to being slaughtered, primarily due to diseases and illnesses [4]. This conclusion was drawn from a 15-month study conducted between 2016 and 2017. Ensuring the health of chickens is achieved by monitoring their behavior. Monitoring chicken behaviors by extracting, and analyzing their trajectories (as shown in Fig. 1), is used as an indicator to ensure their health and well-being. Although the chickens seem fine, sudden death occurs quickly if specific behaviors are displayed, leaving no chance for intervention [5].



(a) Walking trajectory



(b) Eating trajectory



(c) Sleeping trajectory

Fig. 1. Different chicken behavior trajectories.

Different methods for analyzing chicken behavior are utilized (as shown in Fig. 2), one such approach involves using pose estimation as used by Fang et al. [10], where the key points of the chickens are identified and utilized to monitor their behavior. Another method is to analyze the collective behavior of chickens within a farm environment. The manner in which chickens move as a group provides insights into their behaviors. The proposed system (as shown in Fig. 3) primarily concentrates on the third and final approach, which involves tracking the movements of each chicken trajectory individually. The individual actions of the chickens serve as an indicator of their behavior. Fig. 2 shows the previously mentioned methods.

Previous researches that tackled this problem mostly utilized image processing and machine learning to detect and classify chicken behaviors. As illustrated in Table I, Neethira-

TABLE I. PREVIOUS DEEP LEARNING-BASED APPROACHES FOR ANIMAL DETECTION

Algorithm	Application	Dataset Size	Dataset Availability	Accuracy	Reference
Yolov5, Kalman filter	Detecting & Counting	72 Chickens 8 hours x30 FPS	Not available	96%	[6]
Yolov4, Kalman filter	Identify chicken's movement in low light	6 laying hens 1296 frames	Not available	99%	[7]
Yolov5, Detectron2	Quail detection mechanism	5,332 frames	Not available	85.07%,67.15% Average precision	[8]
Yolov5, ResNet18	Trajectory of polar bears	4450 frames	Available	86.4% F1 score	[9]
DeepLabCut and Pre trained ResNet50	Chicken pose estimation and behavior classification	4450 frames	Not available	75% standing,92% preening, 51% walking,96% resting, 62% running,93% eating	[10]
SVM	Prediction of broiler disease	34,280 frames	Not available	97%	[11]
Yolov5,logistic regression ML	Detecting chickens,behaviors	6 hours x30 FPS	Not available	94.7%	Proposed work

jan [6] put forward a system for detecting and counting chickens using Yolov5 and Kalman Filter. Siriani [7] also employed Yolov4 and Kalman filter to monitor chickens' movements in low light conditions. Meanwhile, Fang et al. proposed a system that classifies chicken behavior by analyzing their poses using DeepLabCut [12] and ResNet50 [13].

Table I presents a GAP analysis, which compares the proposed system work with previous studies on animal (chicken) detection and tracking. The table compares the algorithms, applications, data-set sizes, data-set availability, and achieved accuracies used in each previous work. Evaluating the strengths and limitation of each existing approach is achieved by comparing the mentioned factors.

The main contribution of this paper is to investigate the use of chicken's trajectories in order to classify its behavior using classical machine learning models. The proposed system (as shown in Fig. 3) aims to classify three chicken behaviors using their trajectories: walking (as shown in Fig. 1a), eating (as shown in Fig. 1b), and sleeping (as shown in Fig. 1c) (Fig. 4 also shows practical examples of the aforementioned behaviors). This paper utilizes a collected data-set of 90 records split evenly among the three mentioned behaviors. The data-set is to be further explained in the Data Collection section. The proposed model is also evaluated on a benchmarked data-set which is the Human Activity Recognition data-set [14] in order to evaluate its efficiency. Furthermore, a comparison between different classical machine learning models is also conducted in order to determine the highest performing model.

The rest of the paper is organized as follows: Section II discusses the literature review and previous related work, Section III outlines the proposed system's methodology, data collection, and how the system works, Section IV presents the setup and results of the two conducted experiments, Section V discusses the analysis and results produced by the classical machine learning models, Section VI presents the limitation of existing approaches and potential future work ideas, and Section VII states the conclusion of the paper.

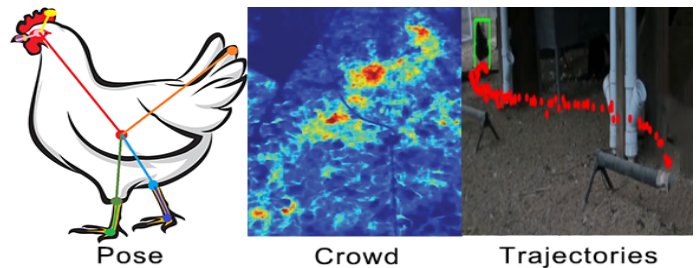


Fig. 2. Methods of chickens behavior classification.

## II. RELATED WORK

In recent years, many studies have focused on detecting and tracking animal behavior with the aim of further understanding the animals and detecting their behavior. The aim of these studies is to develop deep learning models for detecting and tracking animal behaviors. In this section, some of the previous work in this area is presented.

Neethirajan proposed a model which detects chickens, counts them, and extracts each chicken's movement from their bounding box [6]. Neethirajan's model detected chickens from various backgrounds using YOLOv5, and then gave each chicken an ID associated with the bounding box. As a result, the model was able to count and track chickens. To detect each chicken's direction, the proposed system stored the centroid of each bounding box. A challenge this model faced was occlusion, where a chicken disappears in one frame and return in the next frame, resulting in the chicken being given a new ID. To solve this challenge, a Kalman filter was used to compute the distance between each centroid and the old one, in order to check if the chicken had been tracked before.

Siriani et al. proposed a model to detect and track chickens in low light [7]. The proposed model used YOLOv4 in order to detect chickens from input frames in addition to returning the bounding box positions to store in CSV files for the tracking process. Kalman filter was used to predict the next position of each chicken depending on its last positions. This model scored a very high accuracy of 99.9%. However the data-set used was too small in addition to using 10,000 epochs which

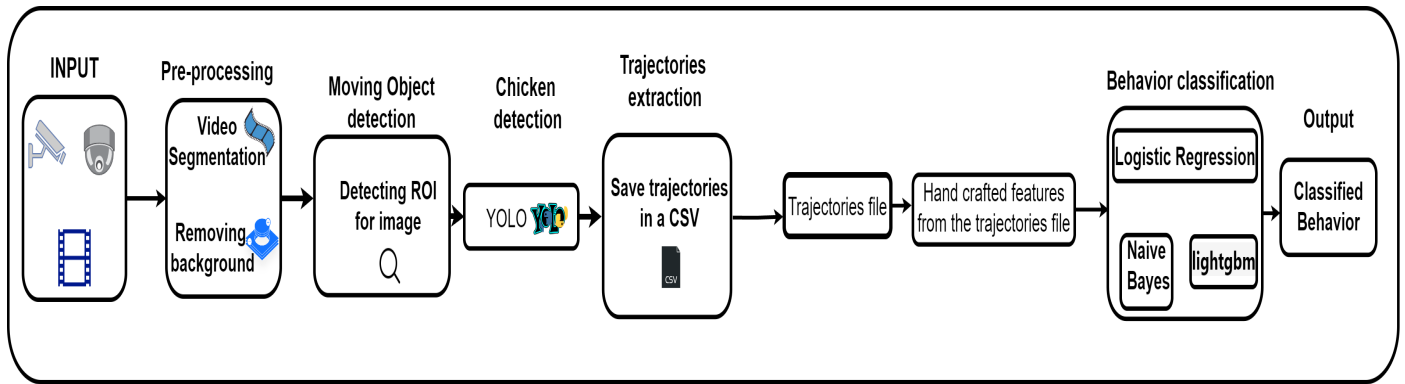


Fig. 3. System architecture.



Fig. 4. Different chickens behaviors.

probably led to over-fitting for their model.

Evangelista et al. proposed a model for detecting quails using YOLOv5 and Detectron2 [8]. The proposed system combined YOLOv5 and Detectron2 to create a faster model for training and validation. However, the model failed to attain a high accuracy and detected touching quails as a single quail.

Zuerl et al. proposed a model for monitoring polar bear behavior using YOLOv5 and ResNet18 [9]. The suggested system consisted of four stages: object detection, classification, coordinate mapping, and analysis and user interface. In the first stage, YOLOv5 was used to detect polar bears. In the second stage, the image was cropped to only include the detected animal, and then ResNet18 was used to classify the animal type. After that, the center of one of the edges in the bounding box was used as the position of the polar bear. Following that, the extracted position was evaluated with respect to the map of the zoo to determine its location. The final stage was the user interface, which displayed the bear's location in the form of a heat map or trajectory, the distance covered in a selected time frame, and the bear's activity. Their model failed to achieve high accuracy. Furthermore, the mapping stage of their approach is difficult to implement, as a map of the farm is required.

The previous work mentioned methods primarily focused on detecting animals and determining their trajectories, without considering any behaviors. In contrast, Fang et al. proposed a model for chicken behavior classification, along with the estimation of chicken poses, using DeepLabCut and a pre-trained ResNet50 model [10]. As the first step, they created a basic skeletal map of the stance of broiler chickens. The ResNet-50 was then trained on the data-set, and DeepLabCut was used to predict the chicken's body position. Afterward,

the chicken's posture was fed into a Naive Bayes model to determine the behavior to which it belongs. However, the limitations of this approach were the low accuracy scores in detecting running and walking behaviors, as the postures for these two classes are similar and difficult to separate.

Okinda et al. presented a distinct approach from the work discussed earlier [11]. This approach employed a support vector machine (SVM) to predict broiler diseases. Unlike the previous methods, this model detects chickens by analyzing images from the perspective of shape representation. To predict the disease, six models were experimented with, and the SVM scored the highest results. However, the model also misclassified some inputs as incorrect days.

However, to date, there has been no published study on behavior detection using chicken trajectories. This paper presents a novel approach for detecting chickens, extracting their trajectories, and classifying those trajectories according to their respective behaviors.

### III. METHODOLOGY

The system proposed in this paper consists of six stages: pre-processing, background subtraction and ROI detection, chicken filtering, trajectory extraction, behavior classification, and output Fig. 3. In this section, a comprehensive explanation of each stage is provided.

#### A. Data Collection

The data-set used in this study was obtained from YouTube videos of chicken barns, totaling 6 hours of video footage. To establish ground truth, the video footage was labeled and trajectories were created with labels such as walking, eating, and sleeping. Each class comprised 30 records, resulting in a total of 90 records. Each record contains the (x, y) positions of the chicken during the recorded clips. The data-set was then split, with 80% used for training and 20% for testing.

For validation purposes, a human activity recognition data-set [15] was used as a benchmark to validate the proposed model. The handcrafted features were calculated from the raw time domain data. The data-set was constructed from 270 records, which contained four behaviors: standing still, walking, jumping, and running, with each class having an equal number of records. The data-set was split in the same manner

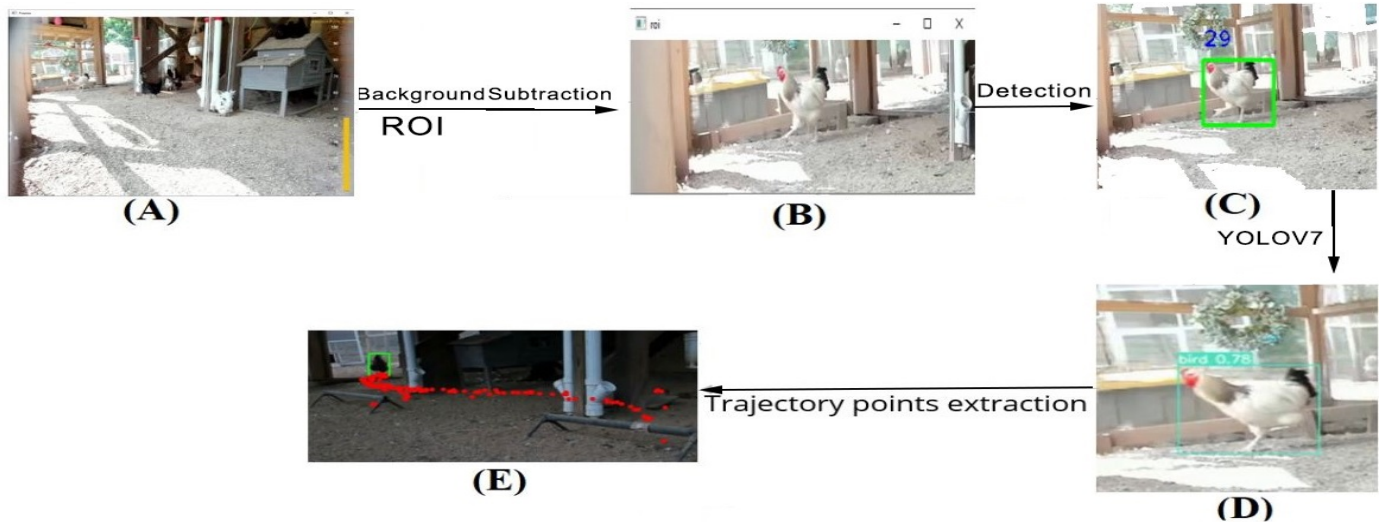


Fig. 5. System stages: (A) Input segmentation, (B) Background subtraction & ROI, (C) Moving object detection, (D) Chicken filtering, (E) Output.

as the constructed data-set, with 80% for training and 20% for testing.

### B. Pre-Processing

The proposed system has a pre-processing stage, which requires video segmentation as its first step. The objective is to analyze the content of the video. To achieve the desired results, the video undergoes processing. The fundamental procedure for processing a video is video segmentation [16]. The input videos are segmented into intervals of 5, 10, and 15 seconds for processing. The results showed that the 10-second segmentation achieved the best outcome, as it was observed that chickens typically transition to a different behavior after 10 seconds.

### C. Background Subtraction and ROI

After video segmentation, the next step is to apply background subtraction to the videos. Background subtraction is commonly used to isolate objects of interest in a scene by comparing an observed image to an estimated version of the image that does not contain any objects of interest [17], [18]. The purpose of applying background subtraction is to determine the trajectory of the chickens as shown in Fig. 5(B).

The process of applying the Region of Interest (ROI) to a moving object is utilized to eliminate any irrelevant data present in the image. In addition to detecting any moving object as demonstrated in Fig. 5(C).

### D. Chicken Filtering

To detect any moving objects. The object detection stage is then performed to determine if the moving object is a chicken. This stage is conducted using YOLO, and if the object detected is a chicken, it is assigned an ID, as shown in Fig. 5(D). Joseph Redmon and others first introduced YOLO in 2016 [19]. By partitioning an image into a grid and producing predictions for each cell in the grid, the algorithm is made to recognise objects in real-time. The distinctive feature of

YOLO is that it only makes one forward pass over the network, enabling real-time image processing. Other object identification methods, on the other hand, call for many forward passes, which makes them slower and less suited for real-time applications. On a number of object detection benchmarks, YOLO has been demonstrated to attain state-of-the-art performance while also being computationally effective. Its widespread application in numerous computer vision tasks is a result of this. In addition to its benefits include quick deep learning network processing times, the capacity to handle larger data-sets, and real-time continuous detection [20], [21].

$$O = \sqrt{(X_{new} - X_{old})^2 + (Y_{new} - Y_{old})^2} \quad (1)$$

Equation 1: Euclidean Distance Equation

The ID of a chicken changes in the case of occlusion and is assigned a new ID [6]. To overcome this challenge, the Euclidean distance is calculated between the chicken's previous location and its new location [22]. If the output of equation 1 is compared to the positions of all existing IDs and found to be less than 150, the chicken is assigned the same ID and the ID's position is updated.

### E. Trajectories Extraction

The first step in trajectories extraction stage is to detect ROI(region of interest) from the image. The ROI is determined by the boundary box added in the previous stage by YOLO.

Next, the location of the chickens is assumed to be at the right edge of the bounding box, and a trajectory point is added every 10 frames. As previously mentioned, processing every frame is a challenging task, and thus trajectory extraction is only performed once every 10 frames [23]. Finally, the trajectories are saved in a CSV (Comma-Separated Values) file to calculate features from it.

$$A = \frac{1}{n} \sum_{i=1}^n a_i = \frac{a_1 + a_2 + \dots + a_n}{n} \quad (2)$$

Equation 2: Mean equation

$$\sigma = \sqrt{\frac{1}{n} \sum_{i=1}^n (x_i - \mu)^2} \quad (3)$$

Equation 3: Standard deviation equation

$$M = \frac{1}{n} \sum_{i=1}^n x_i \quad (4)$$

Equation 4: Median equation

#### F. Hand Crafted features

Since the trajectories are considered time series data [24], several time series features are calculated, such as the mean, median, standard deviation, minimum, and maximum. Mean is used to calculate the average distance covered as shown in equation 2 where  $a_1, a_2, \dots$  and is calculated using the euclidean distance mentioned in equation 1. Median is used to extract the center element from a list of numbers using equation 4. In this case median is used to achieve the medium distance from all distances calculated. Standard deviation is used to determine how close the values are to the average (mean) value. The standard deviation is calculated using the equation 3.

$$DC = \sum_{i=1}^n \sqrt{(X_{new} - X_{old})^2 + (Y_{new} - Y_{old})^2} \quad (5)$$

Equation 5: Distance Covered equation

$$\begin{cases} SS+ = 1 & \text{if } \sqrt{(X_{new} - X_{old})^2 + (Y_{new} - Y_{old})^2} \leq 50 \\ SS+ = 0 & \text{otherwise} \end{cases} \quad (6)$$

Equation 6: Standing still Equation

In addition, some original features were calculated, such as standing still, distance covered, maximum distance covered, and starting-ending. The standing still feature was calculated by subtracting every two consecutive trajectory points, and if the result was less than 50, the chicken is considered to be standing still as shown in equation 6. The distance covered feature was calculated by summing all of the trajectory points as presented in equation 5. The maximum distance covered feature was calculated by determining the maximum difference between each of the consecutive trajectory points. Finally, the starting-ending feature was calculated by subtracting the first trajectory point from the last trajectory point.

After combining and evaluating these features, the best results were achieved by using the following set of features:

standing still, distance covered, mean, maximum distance covered, starting-ending position, and label.

#### G. Behavior Classification

In order to classify the behaviors, three classical machine learning algorithms were used: Logistic Regression, Naive Bayes, and LightGBM.

Classification is the main application of logistic regression. Unlike linear regression, logistic regression data points are not arranged in straight lines. Each group represents a category, and each data point belongs to a specific category. The objective of logistic regression is to find the classification boundary line, which is represented by the regression equation [25], [26]. Because of its distinctive way of expressing uncertain knowledge, its extensive capacity for expressing probability, and its incremental learning features for integrating prior knowledge, the Bayesian technique has quickly become one of the most alluring focal points in many methods. One of the original Bayesian classification algorithms, the Naive Bayesian Classification Algorithm, has a straightforward structure and great processing efficiency. A Naive Bayes classifier has the advantage of only needing to estimate relevant parameters, such as the mean and variance of variables, based on a small sample of training data. The assumption of independent variables means that only the procedure for estimating each variable is required, and that the entire covariance matrix is unnecessary [27], [28].

LightGBM is a new model that is based on the gradient learning framework and decision trees, and implements the concept of boosting. It differs from the XGBoost model in its use of histogram-based methods to accelerate training, conserve memory, and achieve a leaf wise growth approach with depth limitations. The histogram method involves discretizing continuous floating-point eigenvalues into  $k$  bins, which results in a histogram with a width of  $k$ . This algorithm preserves the discretized feature values using 8-bit integers, reducing memory consumption without sacrificing accuracy, without the need for additional storage of pre-sorted results [29], [30].

After processing the input, the extracted features are fed into the model to classify its behavior.

## IV. ANALYSIS AND RESULTS

Two experiments were performed using the same approach that was previously explained. Input videos were divided into 10-second segments. Experiment 1, which focused on chicken behavior, had a sample size of 12 hours footage from a chicken barn, where the selected behaviors were cropped in order to conduct the experiment. Experiment 2 involved the Human Activity Recognition in order to check the efficiency of the proposed system, using the HAR data-set, both experiments also used the same hand crafted features. After pre-processing the segmented videos as shown in Fig. 3, and the trajectories of the chicken were extracted, they were passed to the classical machine learning models in order to determine the highest performing model.

### A. Experiment 1

Experiment 1 aims to classify the three previously mentioned chicken behaviors, while also comparing between different classical machine learning models.

1) *Setup*: The first experiment uses the collected chicken data-set to test the efficiency of the proposed system. It was carried out using 90 records with 30 records allocated for each behavior and split 80-20 between training and testing, respectively. All classical machine learning models were trained and tested on the same data-set, the aforementioned models which are Logistic regression, Naive Bayes, and Light Gradient Boosting Machine (as shown in Fig. 3), were selected in order to determine the best performing model in this particular research.

2) *Results*: The performance of three different machine learning algorithms, namely Naive Bayes, Logistic Regression, and Light Gradient Boosting Machine (LightGBM), were evaluated in this study for the classification of chicken behavior. The results showed that LightGBM outperformed both Naive Bayes and Logistic Regression in terms of accuracy. Results are shown in Table II.

### B. Experiment 2

The second experiment was conducted in order to have the proposed system evaluated on a public and benchmarked data-set.

1) *Setup*: The second experiment makes use of the Human Activity Recognition data-set. It was conducted using 270 records divided evenly between all 4 behaviors which are StandingStill, Walking, Jumping and Running. The records assigned to each behavior was also split 80-20 for training and testing respectively.

2) *Results*: In Experiment 2, the performance of the same three classical machine learning models were evaluated: Logistic Regression, Naive Bayes, and LightGBM. The results, as shown in Table III, indicate that Logistic Regression outperformed both Naive Bayes and LightGBM.

TABLE II. CHICKEN DATASET RESULTS

Model	Accuracy
LightGBM	94.7%
Naive Bayes	91.3%
Logistic regression	89.8%

TABLE III. COMPARING RESULTS WITH PREVIOUS WORK

Refrence	Accuracy
[31]	98.99%
Logistic regression	98.4%
LightGBM	93.9%
Naive Bayes	90.1%

## V. DISCUSSION

### A. Experiment 1

LightGBM performed better than both Naive Bayes and Logistic regression (as shown in Table II) in experiment 1

which was conducted to assess the performance of the proposed system on classifying chicken behavior, this is attributed to several factors. Firstly, LightGBM is a gradient boosting algorithm that builds decision trees sequentially to correct the errors made by previous trees and then work on the leaves of these trees, where each node in the tree is a hand-crafted feature on its own. This results in a more robust model that captures complex and non-linear relationships between the features and the target. Secondly, LightGBM has a tree-pruning mechanism that helps to prevent over-fitting on small data-sets. By removing branches with low information gain, LightGBM is able to reduce the size of the tree and avoid over-fitting, leading to better generalization on unseen data.

On the other hand, Naive Bayes and Logistic Regression are linear models that assume a linear relationship between the features and target, which is not true in this particular experiment, and results in underfitting when the relationship between the features and the target label is more complex. Naive Bayes classifier also considers each of these features to contribute individually in the predicting process, which loses information about how the hand-crafted features are used together to identify the behavior.

### B. Experiment 2

The results of Experiment 2 (as shown in Table III) show that, unlike Experiment 1, Logistic regression scored higher accuracy than both LightGBM and Naive Bayes in the classification of human activities. This improved performance of Logistic Regression compared to Experiment 1 is attributed to the larger data-set size in Experiment 2. In Experiment 1, the small data-set size have resulted in Logistic Regression being underfit, however, with the increased data-set size in Experiment 2, Logistic Regression was able to show higher accuracy. Logistic Regression also robust to outliers and has the ability handle real-world noisy data-sets, which is why it was suitable in this experiment.

The accuracy of both Naive Bayes and LightGBM models stayed around the same percentage (as shown in both Table II, and Table III), due to the same reasons that were mentioned in the discussion section for experiment 1.

## VI. LIMITATIONS AND FUTURE WORK

In this section, the limitations encountered during this research will be outlined, along with potential areas for future exploration for interested researchers.

The paper highlights the potential of using a computer-vision system and machine learning for chicken behavior analysis and surveillance in poultry farms. These technologies can provide valuable insights and serve as useful tools for farmers and researchers. However, there are also several limitations and challenges associated with this approach, which should be taken into consideration. Chickens overlapping: Chickens often move around in groups and can block each others movements (as illustrated in Fig. 6), making it challenging for computer vision algorithms to track and detect individual behaviors. In addition, occlusion can occur when one chicken is partially or completely obstructed from view by another chicken or object in the environment.

Chicken behaviors variability: chickens exhibit a wide range of behaviors, these different behaviors look very similar to each other (as shown in Fig. 1), which further complicates their classification, some of these behaviors are also not feasible to classify using only their trajectories, which means that a future work combining the proposed system with a pose estimation-based system is needed.

Future research to broaden the scope of this study is including the classification of abnormal behaviors observed in poultry farms. Additionally, other machine learning algorithms, larger data-sets, and more complex behavior classification tasks to be explored to further enhance the accuracy and practical applications of the proposed system. The proposed system can also be combined with a pose estimation-based approach to further improve accuracy and widen the classification scope, enabling it to detect different types of behaviors.



Fig. 6. Chickens occlusion.

## VII. CONCLUSION

In conclusion, computer vision and machine learning has the potential to radically transform the animal behavior classification industry providing a reliable and an automated way to monitor and analyze animal behaviors in poultry farms. This study aimed to classify chicken behavior using computer vision and classical machine learning approach.

The proposed system was evaluated on two experiments; the first experiment was conducted using a collected chicken data-set. The second experiment made use of the Human Activity Recognition data-set to test the efficiency of the proposed system on a benchmarked data-set. Three classical machine learning models were given the same training and testing data to determine the best-performing model, also given the same hand crafted features in both experiments. The results showed that the Light Gradient Boosting Machine model outperformed the other models in experiment 1, achieving the highest accuracy of 94.7% in classifying chicken behaviors. While Logistic Regression achieved the highest accuracy of 98.4% in experiment 2 conducted to classify human activities. These findings demonstrate the potential of using machine learning to classify animal behavior and its ability to have practical applications in the poultry industry.

## REFERENCES

- [1] P. He, Z. Chen, H. Yu, K. Hayat, Y. fan He, J. Pan, and H. Lin, "Research progress in the early warning of chicken diseases by monitoring clinical symptoms," *Applied Sciences*, 2022.
- [2] M. A. A. Bakar, P. J. Ker, S. G. H. Tang, H. J. Lee, and B. S. Zainal, "Classification of unhealthy chicken based on chromaticity of the comb," *2022 IEEE International Conference on Computing (ICOCO)*, pp. 1–5, 2022.
- [3] R. Mujawar and P. Jamsandekar, "Data mining in poultry diseases detection: A literature review," vol. 11, 12 2022.
- [4] D. Cbeneau, "The number of chickens who die before reaching the kill blade," Jul 2022. [Online]. Available: <https://animalequality.org/blog/2019/11/07/the-shocking-number-of-chickens-who-die-before-reaching-the-kill-blade/>
- [5] R. C. Newberry, E. E. Gardiner, and J. R. Hunt, "Behavior of chickens prior to death from sudden death syndrome." *Poultry science*, vol. 66 9, pp. 1446–50, 1987.
- [6] S. Neethirajan, "Chicktrack – a quantitative tracking tool for measuring chicken activity," *Measurement*, vol. 191, p. 110819, 2022. [Online]. Available: <https://www.sciencedirect.com/science/article/pii/S0263224122001154>
- [7] A. Siriani, V. Kodaira, S. Mehdizadeh, I. Nääs, D. Moura, and D. Florentino Pereira, "Detection and tracking of chickens in low-light images using yolo network and kalman filter," *Neural Computing and Applications*, vol. 34, 08 2022.
- [8] I. R. Evangelista, L. Catajay, M. G. Palconit, M. G. A. Bautista, R. Concepcion II, E. Sybingco, A. Bandala, and E. Dadios, "Detection of japanese quails ( *coturnix japonica* ) in poultry farms using yolov5 and detron2 faster r-cnn," *Journal of Advanced Computational Intelligence and Intelligent Informatics*, vol. 26, pp. 930–936, 11 2022.
- [9] M. Zuerl, P. Stoll, I. Brehm, R. Raab, D. Zanca, S. Kabri, J. Happold, H. Nille, K. Prechtel, S. Wuensch, M. Krause, S. Seegerer, L. von Fersen, and B. Eskofier, "Automated video-based analysis framework for behavior monitoring of individual animals in zoos using deep learning—a study on polar bears," *Animals*, vol. 12, no. 6, 2022. [Online]. Available: <https://www.mdpi.com/2076-2615/12/6/692>
- [10] C. Fang, T. Zhang, H. Zheng, J. Huang, and K. Cuan, "Pose estimation and behavior classification of broiler chickens based on deep neural networks," *Computers and Electronics in Agriculture*, vol. 180, p. 105863, 2021. [Online]. Available: <https://www.sciencedirect.com/science/article/pii/S0168169920305159>
- [11] C. Okinda, M. Lu, L. Liu, I. Nyalala, C. Muneri, J. Wang, H. Zhang, and M. Shen, "A machine vision system for early detection and prediction of sick birds: A broiler chicken model," *Biosystems Engineering*, vol. 188, pp. 229–242, 2019. [Online]. Available: <https://www.sciencedirect.com/science/article/pii/S1537511019308438>
- [12] A. Mathis, P. Mamidanna, K. M. Cury, T. Abe, V. N. Murthy, M. W. Mathis, and M. Bethge, "DeepLabcut: markerless pose estimation of user-defined body parts with deep learning," *Nature Neuroscience*, vol. 21, pp. 1281–1289, 2018.
- [13] B. Koonce, *ResNet 50*. Berkeley, CA: Apress, 2021, pp. 63–72. [Online]. Available: [https://doi.org/10.1007/978-1-4842-6168-2\\_6](https://doi.org/10.1007/978-1-4842-6168-2_6)
- [14] A. A. Nahid, N. Sikder, and I. Rafi, "Ku-: An open dataset for human activity recognition," 2020.
- [15] A.-A. Nahid, N. Sikder, and I. Rafi, "Ku-har: An open dataset for human activity recognition," *Mendeley Data*, vol. 3, 2020.
- [16] T. Zhou, F. Porikli, D. J. Crandall, L. V. Gool, and W. Wang, "A survey on deep learning technique for video segmentation," *IEEE Transactions on Pattern Analysis and Machine Intelligence*, pp. 1–20, 2022.
- [17] T. Bouwmans and B. García García, "Background subtraction in real applications: Challenges, current models and future directions," 01 2019.
- [18] S. Agrawal, "Backgrounds subtrcation," 03 2020.
- [19] J. Redmon and A. Farhadi, "Yolo9000: better, faster, stronger," *arXiv preprint arXiv:1612.08242*, 2017.
- [20] Y. Fang, X. Guo, K. Chen, Z. Zhou, and Q. Ye, "Accurate and automated detection of surface knots on sawn timbers using yolo-v5 model," *BioResources*, vol. 16, no. 3, p. 5390–5406, 2021.
- [21] A. Kuznetsova, T. Maleva, and V. Soloviev, *YOLOv5 versus YOLOv3 for Apple Detection*. Cham: Springer International Publishing, 2021, pp. 349–358. [Online]. Available: [https://doi.org/10.1007/978-3-030-66077-2\\_28](https://doi.org/10.1007/978-3-030-66077-2_28)
- [22] B. O'Neill, "Chapter 2 - frame fields," in *Elementary Differential Geometry (Second Edition)*, second edition ed., B. O'Neill, Ed.

- Boston: Academic Press, 2006, pp. 43–99. [Online]. Available: <https://www.sciencedirect.com/science/article/pii/B9780120887354500067>
- [23] H. Luo, W. Xie, X. Wang, and W. Zeng, “Detect or track: Towards cost-effective video object detection/tracking,” *Proceedings of the AAAI Conference on Artificial Intelligence*, vol. 33, no. 01, pp. 8803–8810, Jul. 2019. [Online]. Available: <https://ojs.aaai.org/index.php/AAAI/article/view/4906>
- [24] N. Vaughan and B. Gabrys, “Comparing and combining time series trajectories using dynamic time warping,” *Procedia Computer Science*, vol. 96, pp. 465–474, 2016, knowledge-Based and Intelligent Information & Engineering Systems: Proceedings of the 20th International Conference KES-2016. [Online]. Available: <https://www.sciencedirect.com/science/article/pii/S187705091631907X>
- [25] R. Couronné, P. Probst, and A.-L. Boulesteix, “Random forest versus logistic regression: A large-scale benchmark experiment,” *BMC Bioinformatics*, vol. 19, no. 1, 2018.
- [26] X. Zou, Y. Hu, Z. Tian, and K. Shen, “Logistic regression model optimization and case analysis,” in *2019 IEEE 7th International Conference on Computer Science and Network Technology (ICCSNT)*, 2019, pp. 135–139.
- [27] E. Aditya, Z. Situmorang, B. H. Hayadi, M. Zarlis, and Wanayumini, “New student prediction using algorithm naive bayes and regression analysis in universitas potensi utama,” in *2022 4th International Conference on Cybernetics and Intelligent System (ICORIS)*, 2022, pp. 1–6.
- [28] H. Chen, S. Hu, R. Hua, and X. Zhao, “Improved naive bayes classification algorithm for traffic risk management,” *EURASIP Journal on Advances in Signal Processing*, vol. 2021, no. 1, 2021.
- [29] J. Fan, X. Ma, L. Wu, F. Zhang, X. Yu, and W. Zeng, “Light gradient boosting machine: An efficient soft computing model for estimating daily reference evapotranspiration with local and external meteorological data,” *Agricultural Water Management*, 2019.
- [30] G. Ke, Q. Meng, T. Finley, T. Wang, W. Chen, W. Ma, Q. Ye, and T.-Y. Liu, “Lightgbm: A highly efficient gradient boosting decision tree,” in *NIPS*, 2017.
- [31] N. Sikder, M. A. R. Ahad, and A. Nahid, “Human action recognition based on a sequential deep learning model,” 10 2021.