Received August 15, 2019, accepted October 9, 2019, date of publication November 11, 2019, date of current version December 17, 2019.

Digital Object Identifier 10.1109/ACCESS.2019.2952912

Efficient Herd – Outlier Detection in Livestock Monitoring System Based on Density – Based Spatial Clustering

ZOOL HILMI ISMAIL[1], (Senior Member, IEEE), ALAN KH'NG KEAN CHUN[2], AND MOHD IBRAHIM SHAPIAI RAZAK[2], (Member, IEEE)

[1]Centre for Artificial Intelligence and Robotics, Universiti Teknologi Malaysia, Kuala Lumpur 54100, Malaysia
[2]Malaysia-Japan International Institute of Technology, Universiti Teknologi Malaysia, Kuala Lumpur 54100, Malaysia

Corresponding author: Zool Hilmi Ismail (zool@utm.my)

This work was supported in part by the Malaysia-Japan International Institute of Technology under Grant Q.J130000.2509.05H54, and in part by the Universiti Teknologi Malaysia under Grant R.K130000.7843.4J276 and Grant Q.K130000.2543.19H21.

ABSTRACT In today's society, increasing the quality and the productivity of dairy products are very important and need detailed data collection and analysis. Manual collection of data and its analysis for livestock monitoring is costly in terms of high man power and time consumption. In order to overcome this deficit, object detection and clustering methods are investigated in this research as it is in line with Smart Farming 4.0. Faster RCNN is used to help ranchers to detect livestock while clustering methods help to detect the herds and outliers effectively and efficiently. In clustering methods, K-means clustering technique and Density-Based Spatial Clustering of Application with Noise or DBScan clustering technique are adopted. In K-means clustering, k is an important parameter which represents the number of clusters. By changing the number of clusters, the pattern of clusters is observed. Then, the best k value is selected. In DBScan clustering, epsilon is an important parameter which represents the circle radius from a particular data point. The higher the value of epsilon, the formation of clusters becomes easier as it is easy to accept data point in a larger circle radius to form cluster. By changing the epsilon, the pattern of cluster is observed and chosen. Euclidean distance and Manhattan distance are used to compare the effects of different distance metrics on the results of clusters. Cluster pattern is compared between K-means and DBScan techniques. Obtained results show that DBScan overwhelmed K-means in term of efficient clustering in detecting the herds and outliers of livestock.

INDEX TERMS Smart farming, artificial intelligence, livestock monitoring system, region-CNN.

I. INTRODUCTION

In agriculture industry, livestock monitoring and management affects the productivity and quality of dairy products. Livestock monitoring refers to the inspection of livestock to trace the activity levels, health, foods and waters intake level and social interaction status [1]. Livestock monitoring involves the process of collecting data from the livestock and analyze it. This process directly affects the decision of farmers on livestock management. Therefore, livestock monitoring is a crucial process for production of dairy product. Conventional monitoring methods are done manually. Each individual livestock is labeled by ear tag, ear notch, tattoo, or marks. This helps farmers to identify the individual livestock. During the inspection of livestock, the activity levels, food and water intake levels and health conditions are very difficult to acquire as it is biological data that cannot be measured and it is unpredictable [2].

In addition, the man power and time consumption of manual inspection is very costly. In order to increase the productivity, sustainability and the quality of livestock farming industries, an efficient and effective system is implemented to improve the livestock monitoring process and ease the farmers from doing analysis manually. Recently, many sensors such as wearable devices are introduced as ear tag device with sensors is used to obtain and record essential information of the individual livestock such as heart rate, stress level and blood pressure [3]. Some of the wearable devices consist of

The associate editor coordinating the review of this manuscript and approving it for publication was el-Hadi M. Aggoune.

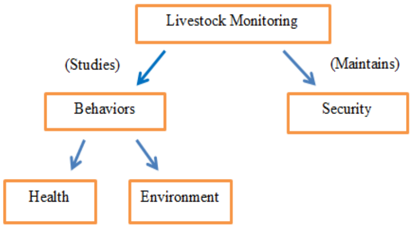

FIGURE 1. Role of livestock monitoring in agricultural industrial [3].

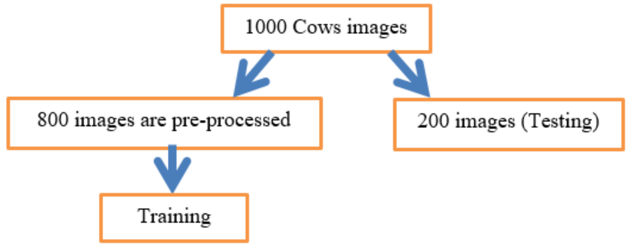

FIGURE 2. Dataset for building a cow detection model.

Global Positioning System (GPS) which helps to track the location of livestock. However, installation and maintenance of GPS in the wearable devices for every individual livestock are very expensive. Therefore, object detection using artificial intelligence is an alternative method that can track the location of livestock.

Object detection is a deep learning method to locate and identify objects in an image through feature extraction and learning algorithm [4]. Nowadays, object detection is getting more popular as it uses computer vision to process large amount of data to identify objects in images. This technique can be applied in livestock monitoring to detect the cows. Clustering is a machine learning method that uses grouping process to divide data into numbers of clusters based on the similarity of data. It is commonly used in many applications such as data mining, outlier detection, pattern recognition [5]. The outputs of object detection are post processed using clustering to detect the herds and outliers. Clustering is used in this study to detect the herds and outliers of livestock and two most popular methods are being compared between K-means and DBScan application.

In order to expand the productivity and quality of livestock product, the livestock monitoring process is a necessity. It affects the decisions of farmers on livestock management. For instance, isolation can be made to prevent the spreading of diseases if the sickness of cows is detected at early stage. However, livestock monitoring is not an easy task as it requires a lot of efforts to collect data and analyze it to study the behaviors of livestock. The data of monitoring are mostly biological inherently variables which are unpredictable [6]. This supports the inefficiency of manual data collection and its analysis. However, monitoring livestock is crucial as it provides information of livestock to the farmers as in Fig. 1 which shows the role of livestock monitoring system.

It is interesting to note that livestock monitoring is commonly used to study the behaviors of livestock that disclose the health conditions of individual livestock and interaction between livestock and its environment while maintaining the security of the system. It prevents livestock theft and livestock lost. In this study, the aim is to detect the herds and outliers to study the herd patterns that reflect the behavior of livestock. In K-means clustering, k is an important parameter which represents the number of clusters while in DBScan clustering, epsilon is an important parameter which represents the circle radius from a particular data point. The higher the value of epsilon, the formation of clusters becomes easier as it is easy to accept data point in a larger circle radius to form cluster. Euclidean distance and Manhattan distance are used to compare the effects of different distance metrics on the results of clusters. Cluster pattern is compared between K-means and DBScan techniques. Experimental results on a group of livestock are presented to demonstrate the effectiveness of the clustering in detecting the herds and outliers.The rest of the paper is organized as follow: Section II describes on the approach while Section III presents the data extraction. Section IV discusses on data clustering along with its analysis in different number of clusters. The results are shown in Section V. Finally, the summary and conclusions are presented in Section VI.

II. APPROACH

This work uses object detection and clustering method in monitoring livestock. Therefore, many cow images are required to be collected as image datasets. Estimated around 1000 images of cows are obtained from the farm compounds in East Malaysia. The images were captured by a drone about 60 meters from the ground. Each image contains more than 1 cow from the top view. Object detection model is applied to detect and identify cows only. A large number of images of cows are required. Fig. 2 shows that out of 1000 images of cows in object detection, 800 images are pre-processed before training and 200 images are used for testing.

A. DATA PRE-PROCESSING

Data pre-processing is a process that modify the image dataset before the images are fed into Faster RCNN for training. Many CNN models expect square shape images as input data. Therefore, resizing or cropping of images is needed before feeding into Faster RCNN. In addition, Red, Green, and Blue (RGB) channels are reduced into single grayscale channel because it saves time for training of Faster RCNN [8]. Fig. 3 shows the pre-processing of input images.

Originally, the images are in RGB color and have a dimension of 8688 $\times$ 5792 pixels. At pre-processing stage, the images are then decolorized into single grayscale channel and resized to 105 $\times$ 105 pixels. Then, the images are labeled manually with bounding boxes using a labeling tool (Labelbox).

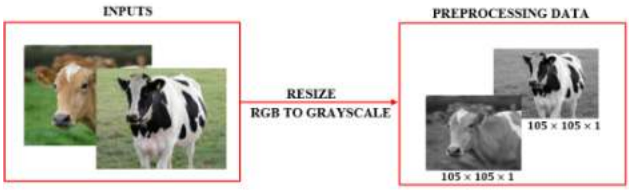

FIGURE 3. Process of building object detection model [8].

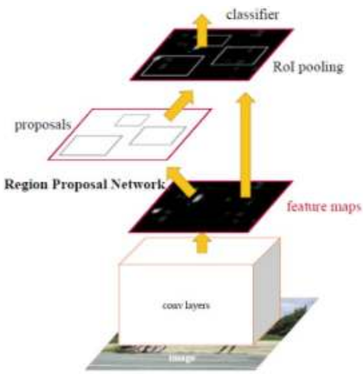

FIGURE 4. Faster RCNN architecture [9].

B. FASTER REGION CONVOLUTIONAL NEURAL NETWORK

Faster Region Convolutional Neural Network or Faster RCNN is a deep learning algorithm widely used in object detection. It consists of two parts. The two parts are Region Proposal Network (RPN) and detection network. Region proposal network is used to generate region proposal while detection network is used to detect objects. Both of these are executed in the same convolutional network. In Fast RCNN and RCNN, region proposals are generated by selective search before CNN is used to do classification of objects and detection of bounding boxes. In Faster RCNN, region proposal generation is done using CNN instead of using selective search. This speeds up object detection process because CNN is shared between RPN and detection network in Faster RCNN architecture.

In RPN, a sliding window is used for locations on the feature maps. In the default configuration of Faster RCNN, 9 anchor boxes are used for each location to generate region proposals [9]. The number of anchors depends on the size of width × height feature map. The number of anchors equals to width × height × 9. Then, locations in feature map are pre-checked in RPN network to identify the location that contains objects. The corresponding locations and bounding boxes are sent to detection network for object detection.

III. DATA EXTRACTION

In this stage, the output data from object detection are extracted before feeding into clustering process. Each bounding box is converted into dot for clustering purpose. In order to convert the bounding box into dot, the coordinates of the centroid of each bounding box are needed as in Fig. 5.

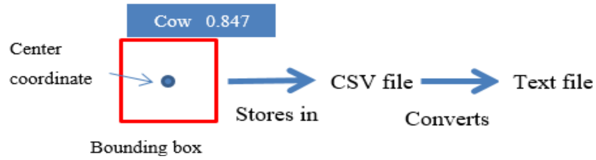

FIGURE 5. Process flow of data extraction.

```
if save_img:
    plt.savefig(im_file, bbox_inches='tight')
X_C=np.array(X_C):Y_C=n.array(Y_C)
mean=np.column_stack((X_C, Y_C))
np.savetxt("/media/live_stock/data/out2/out{}.csv",format(index),
mean, delimiter=",")
```

FIGURE 6. Part of code to data extraction from bounding boxes.

	A	B
1	6.72E+02	1.62E+03
2	6.89E+02	1.28E+03
3	9.77E+02	1.69E+03
4	9.89E+02	1.46E+03
5	2.79E+03	1.95E+03
6	7.76E+02	2.43E+02
7	2.52E+03	1.35E+03
8	4.42E+02	6.06E+01
9	8.56E+02	1.48E+03
10	5.62E+02	1.07E+03
11	2.14E+03	1.03E+03
12	2.35E+03	8.66E+02
13	3.29E+03	8.10E+02
14	8.74E+02	9.19E+02
15	6.40E+02	1.12E+03
16	1.37E+03	1.90E+03
17	2.59E+03	1.36E+03
18	1.62E+03	1.83E+03

FIGURE 7. Dataset in CSV file.

First, center coordinates of the output data from object detection are extracted and stored in a CSV file. The process of storing center coordinates of bounding boxes into a CSV file is done using python language. The code and CSV file dataset is shown in Fig. 6 and Fig. 7, respectively.

Column A represents coordinates in x-axis and column B represents coordinates in y-axis. Each row represents a center coordinate of bounding box. There are 53 center coordinates in total.

IV. DATA CLUSTERING

Data clustering is a process of partitioning data set into small groups. It is commonly used in applications for pattern recognition, data mining and outlier detections. There are many types of clustering algorithms including partitioning clustering, density-based clustering, model-based clustering, hierarchical clustering and so on as shown in Table 1 below.

In this work, a partitioning clustering method and a density-based clustering method are selected. This is because partitioning clustering and density-based clustering are popular and their data computation approaches are simple and efficient in determining the patterns of clusters and detecting outliers. The patterns of the results are observed and compared on partitioning clustering and density-based clustering. There are many types of partitioning clustering method and

TABLE 1. Summary on clustering algorithms.

Proposed method	Advantages
Model-based clustering: mixture-model approach [5]	Use of Bayes factors to compare models is allowed. Systematic approach to select parameterization of model.
Hierarchical clustering: Agglomerative approach [10]	Only a similarity measure is needed. Input parameter is not needed.
Partitioning clustering: K-means approach [11]	Easy to implement and understand. Fast computation. Good robustness of cluster due to exclusive clustering.
Density-based Clustering: DBScan approach [12]	Able to detect outliers. Number of clusters does not need to be defined in advance. Able to find non-spherical shaped clusters.

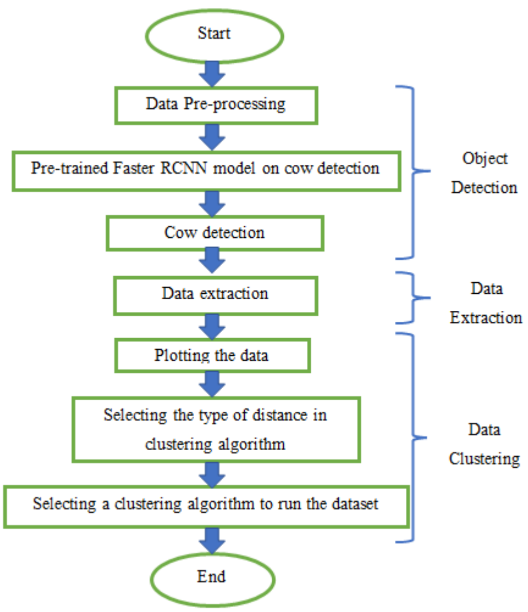

FIGURE 8. Overview of the system flow [13].

density-based clustering method. In this research, K-means clustering algorithms is selected for partitioning clustering method and DBScan clustering algorithms is chosen for density-based clustering method. There are three steps in data clustering process which are:

i. Create a scatter plot from the input data in text file.
ii. Choosing a clustering algorithm.
iii. Selecting a type of distance metric for the chosen clustering algorithm.

In clustering, computation depends on the type of distance metric. Two distance metrics are used to see the results of different distance metric on detecting herds and outliers. The distance metrics are Euclidean distance and Manhattan distance.

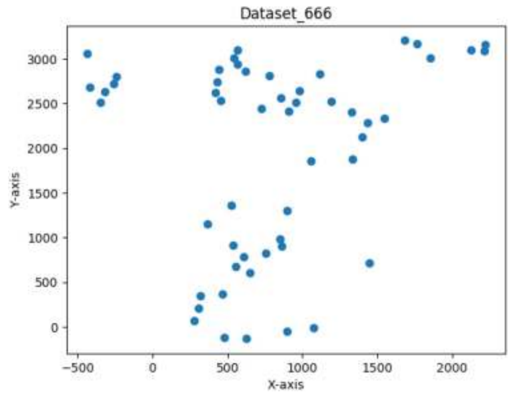

FIGURE 9. Scatter plot of dataset.

A. SCATTER PLOT

One way to investigate the shape of the experimental design is to plot the raw data [14]. The first step in data clustering is to create a plot using Matplotlib library. It provides function of plotting graph. After creating an empty plot, the dataset from text file is loaded into the plot. Then, each coordinate from the text file is plotted on the plot to form a scatter plot where each dot represents a coordinate of input data. Fig. 9 shows the scatter plot after loading the data from the text file. There are 53 dots in total. Each dot represents a center coordinate of bounding box.

B. EUCLIDEAN DISTANCE

Euclidean distance describes the direct distance between two points in Euclidean space. Euclidean distance is commonly used as the default distance metric in many clustering methods such as K-means algorithm. Euclidean distance uses Pythagorean Theorem to calculate the direct distance between points [15]. The way of calculating Euclidean distance is shown as follows

$$dist_E = \sqrt{(x_1 - x_2)^2 + (y_1 - y_2)^2} \tag{1}$$

where x_1, and x_2 represent the coordinates of point A and point B in X-axis, respectively. Meanwhile y_1, and y_2 represent the coordinates of point A and point B in Y-axis, respectively. It is commonly used in clustering because it is simple to compute in 2-Dimensional (2D) and 3-Dimensional (3D) data. Fig. 10 provide a better understanding of Euclidean distance and its computation. The blue line which shows the shortest distance line from point A to point B is the Euclidean distance.

C. MANHATTAN DISTANCE

Manhattan distance is the path distance between two points in a grid [16]. The equation of Manhattan distance is simpler than the Euclidean distance because it is the absolute sum of difference between two points on Cartesian plane as given in the following function

$$dist_M = |(x_1 - x_2) + (y_1 - y_2)| \tag{2}$$

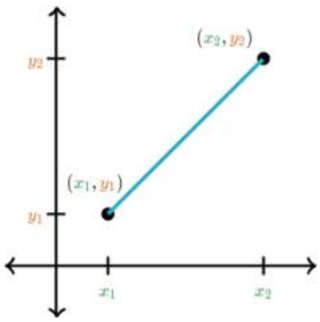

FIGURE 10. Euclidean distance example [15].

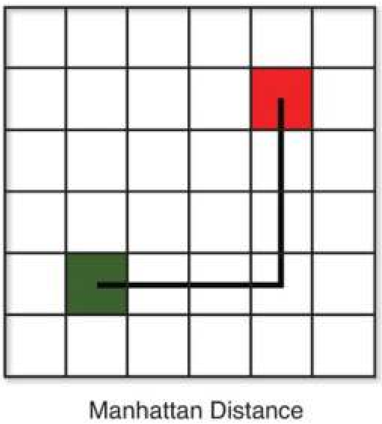

FIGURE 11. Manhattan distance example [16].

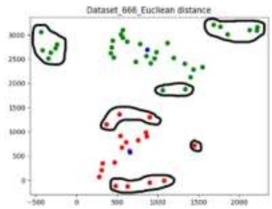

FIGURE 12. K-means with k=2.

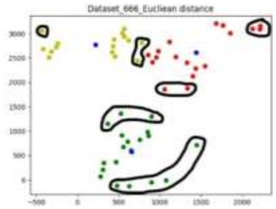

FIGURE 13. K-means with k=3.

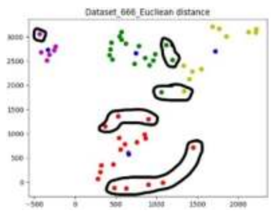

FIGURE 14. K-means with k=4.

where x_1, x_2, y_1, and y_2 are previously defined in (1). Manhattan distance is used in Cartesian plane because it provides a grid that aids the computation of it for human.

D. K-MEANS CLUSTERING

There are many partitioning clustering techniques and one of the most popular is K-means clustering [17]. K-means clustering only works when the number of clusters is set. This means that it is not suitable to be applied in applications that require it to detect the number of clusters in a given dataset. In K-means clustering, k value is automatically partitioning a data set into groups which is an important parameter that represents the number of clusters which the centroids are randomly assigned on the scatter plot [18]. In clustering assignment stage, it computes the distance metric for each remaining data point with each centroid and assigned the data point to the nearest centroid. Thus, it makes k clusters. In moving centroid stage, it re-computes the centroid of each cluster by taking mean of all data points in a particular cluster. It reassigns each data points with respect to new centroid. These steps continue to repeat until convergence is attained.

E. DBScan CLUSTERING

There are many density-based clustering techniques. DBScan is selected out of other the density-based clustering methods. This is because DBScan is a simple clustering algorithm that is good at estimating the number of clusters and detecting outliers. There are two parameters in DBScan algorithm. The parameters are epsilon and minPoint. Epsilon is the minimum distance between two points to measure similarity of spatial data while minPoint is the minimum number of points to form a dense region [19]. In the beginning, epsilon and minPoint are set. After setting the parameters, an arbitrary point is picked in the dataset. If the neighbor points of the particular core point are less than the minPoint within the distance of epsilon and not a part of any other cluster, then these points are considered as outliers. If the neighbor points are more than the minPoint within the distance of epsilon from that point, it forms a cluster. Then, the cluster slowly expands by checking all the points within the epsilon distance around the core points to check their minPoint within the epsilon distance. The cluster grows slowly until no points to add to the cluster [20]. Next unvisited arbitrary point is picked to form a new cluster. The process keeps repeating until all the points are into clusters.

V. RESULTS AND DISCUSSION

There are five experiments in total. These five experiments show the results on K-means clustering with different number of clusters, DBScan with different epsilon value, Euclidean distance versus Manhattan distance in K-means clustering, Euclidean distance versus Manhattan distance in DBScan clustering and K-means clustering versus DBScan clustering

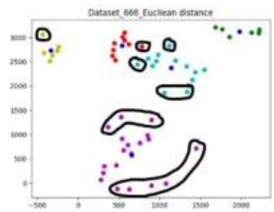

FIGURE 15. K-means with k=5.

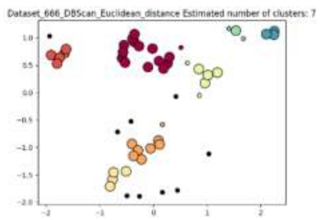

FIGURE 16. DBScan epsilon=0.3.

TABLE 2. Number of outliers detected in K-means clustering with different k value.

K-means clustering with Euclidean distance	
No. of Clusters (unit)	No. of outliers (unit)
2	22
3	16
4	13
5	14

respectively. The dataset contains 53 data points and each data point represents a cow.

A. EXPERIMENT 1

In experiment 1, K-means clustering with different parameter values is done. The parameter represents k value, which also known as the number of clusters. Each colour represents a cluster or a herd. The blue dots represent the centroids of cluster. The dots inside the circles represent the outliers while the dots outside the circles represent the herds.

The table indicates that the number of outliers increases when the number of clusters increases. This is because the distribution of the dataset becomes better in shapes when there are more clusters assigned to the small herds in the plot. The lowest number of clusters scores the highest number of outliers on this dataset in K-means clustering. K-means clustering with four clusters outputs gets the least outliers hence it is deducted that the most suitable number of clusters for this dataset is 4.

B. EXPERIMENT 2

In experiment 2, DBScan clustering with different parameter values is done. The parameter in DBScan clustering represents epsilon value. Each colour represents a cluster or a herd. The black dots represent the outliers.

The result shows that the number of clusters and the number outliers decrease when the value of epsilon increases.

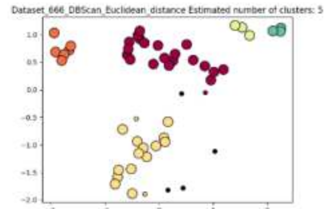

FIGURE 17. DBScan epsilon=0.4.

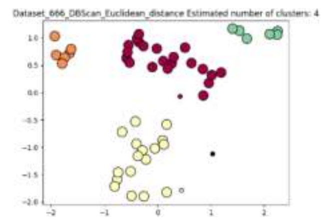

FIGURE 18. DBScan epsilon=0.5.

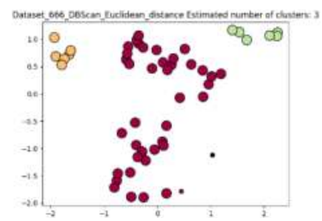

FIGURE 19. DBScan epsilon=0.6.

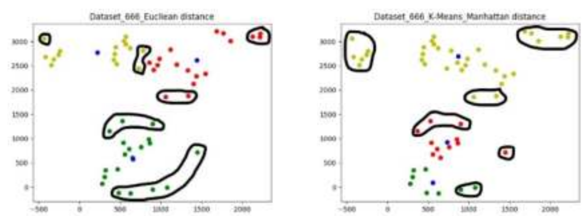

FIGURE 20. Euclidean distance versus manhattan distance with 3 centroids in K-means.

TABLE 3. Number of clusters and outliers obtained in DBScan with different epsilons.

DBScan clustering using Euclidean distance and minPoint = 3		
Epsilon (unit)	No. of clusters generated	No. of outliers
0.3	7	9
0.4	5	4
0.5	4	1
0.6	3	1

As the value of epsilon increases, the chances of getting data points within the epsilon around the core points more than 3 increases. Therefore, it becomes easier for data points to merge to form a large cluster. This results in the decline in the number of clusters and the number of outliers. However, it does not mean that the epsilon that gets the least number of

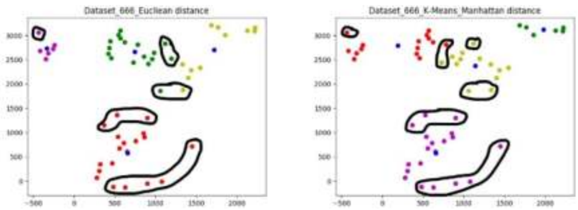

FIGURE 21. Euclidean distance versus manhattan distance with 4 centroids in K-means.

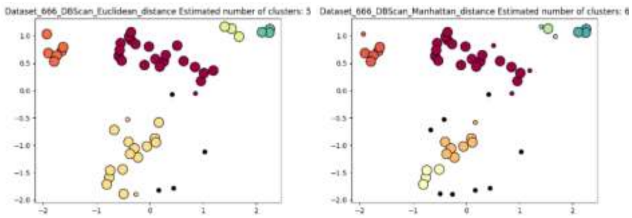

FIGURE 22. Euclidean distance versus Manhattan distance with epsilon=0.4 in DBScan.

TABLE 4. Number of outliers using euclidean distance versus manhattan distance.

	No. of outlier	
	Euclidean distance	Manhattan distance
No. of centroid = 3	16	20
No. of centroid = 4	13	14

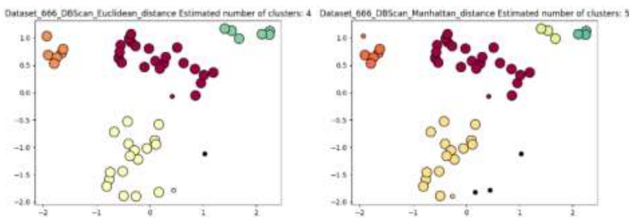

FIGURE 23. Euclidean distance versus manhattan distance with epsilon=0.5 in DBScan.

outliers is the most suitable parameter value for this dataset. It shows that when epsilon equals to 0.4, it outputs a better herd patterns in the dataset compared to that of using epsilon equals to 0.3. When epsilon is equal to 0.3, many data points are detected as outliers when some of the data points are supposed to be clustered.

C. EXPERIMENT 3

In experiment 3, K-means clustering with Euclidean distance and Manhattan distance are done respectively. The blue dots represent the centroids of cluster. The dots inside the circles represent the outliers while the dots outside the circles represent the herds as shown in Fig. 20 and 21.

It indicates that the number of outliers detected in Manhattan distance is more than that in Euclidean distance. Theoretically, clustering using Euclidean distance and Manhattan distance have no much difference. However, Euclidean distance performs better than the Manhattan distance in terms of the pattern of cluster when the number of clusters is three. The result of Manhattan distance shows that two small herds on the top of the graph are too far away from its cluster centroids. This leads to higher number of outliers detected compared to the cluster pattern of Euclidean distance. Therefore, it is said that Euclidean distance fits more in this dataset to detect the herds and outliers.

TABLE 5. Number of outliers using euclidean distance versus manhattan distance.

	Euclidean distance		Manhattan distance	
	No. of cluster	No. of outlier	No. of cluster	No. of outlier
epsilon=0.4	5	4	6	7
epsilon=0.5	4	1	5	3

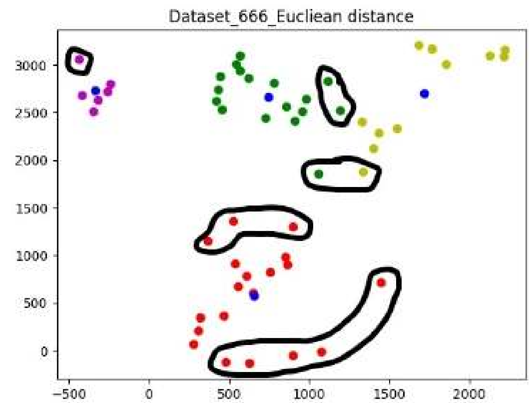

FIGURE 24. K-means with k=4 using euclidean distance.

D. EXPERIMENT 4

In experiment 4, DBScan clustering with Euclidean distance and Manhattan distance are done respectively.

The results show that the number of outlier and the number of clusters in Manhattan distance is more than that in Euclidean distance. The cluster pattern of DBScan clustering using Manhattan distance is better than that of DBScan clustering using Euclidean distance. Using Manhattan distance in DBScan clustering, the patterns of herds are better in shape and the number of outliers is more precise. Therefore, DBScan clustering with Manhattan distance fits more in this dataset.

E. EXPERIMENT 5

In experiment 5, DBScan clustering with Euclidean distance and K-means clustering with Euclidean distance are done respectively. The parameter in K-means clustering represents k value. K value is set to 4. The parameter in DBScan clustering represents epsilon value. Epsilon value is set to 0.4.

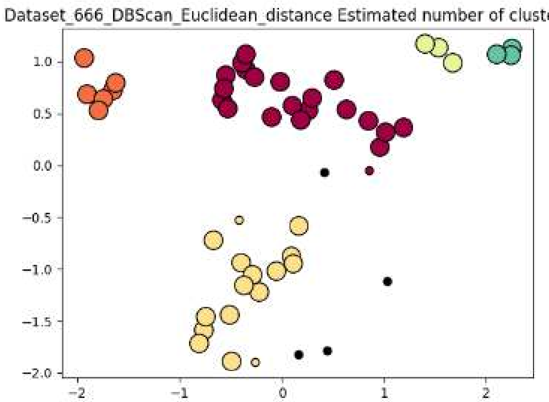

FIGURE 25. **DBScan with epsilon=0.4 using euclidean distance.**

Fig. 24 and 25 show that the K-means algorithm is more sensitive than DBScan algorithm regarding outlier detection. However, the shape of cluster in DBScan is better than the shape of cluster in K-means. The yellow coloured cluster in K-means algorithm shows that the data points are very far away from its cluster centroid. Therefore, DBScan is better than K-means clustering in terms of detecting herds and outliers in this dataset.

VI. CONCLUSION

In this paper, a cow detection system is presented using a pre-trained Faster RCNN network with Caffe framework. Then, the grouping of herds is briefly proposed using clustering method on the output of cow detection. Note that K-means clustering and DBScan clustering are used to determine which clustering method is better in detecting the herds and outliers. Besides, to analyze the effect of clustering using different type of distance metrics, Euclidean distance and Manhattan distance are used in this study. From the experimental results, DBScan clustering is better than K-means clustering in terms of herd pattern and outlier detection in livestock monitoring. In K-means clustering, Euclidean distance produce a better grouping of herds. However, in DBScan clustering, Manhattan distance produces a better herd and outlier detection. It is deducted that the type of distance metric causes the detection of herds and outliers to perform differently in different clustering method. In conclusion, DBScan clustering is more suitable to be implemented in livestock monitoring for local ranchers in line with Smart Farming 4.0.

REFERENCES

[1] M. C. T. Scholten, I. J. M. de Boer, B. Gremmen, and C. Lokhorst, "Livestock farming with care: Towards sustainable production of animal-source food," *NJAS-Wageningen J. Life Sci.*, vol. 66, pp. 3–5, Nov. 2013.

[2] H. Erden, I. A. Kaya, and E. Çama ircio lu, "Livestock monitoring system," in *Proc. 4th Int. Conf. Agro-Geoinform. (Agro-Geoinform.)*, 2015, pp. 73–76.

[3] J. Foulkes, P. Tucker, M. Caronan, R. Curtis, L. G. Parker, C. Farnell, B. Sparkman, G. Zhou, S. C. Smith, and J. Wu, "Livestock management system," in *Proc. Int. Conf. Embedded Syst. Appl.*, 2013, pp. 1–7.

[4] Y. Amit, *2D Object Detection and Recognition: Models, Algorithms and Networks*. Cambridge, MA, USA: MIT Press, 2002, pp. 21–34.

[5] D. Sonagara and S. Badheka, "Comparison of basic clustering algorithms," *Int. J. Comput. Sci. Mobile Comput.*, vol. 3, no. 10, pp. 58–61, 2014.

[6] T. J. DeVries, M. A. G. von Keyserlingk, D. M. Weary, and K. A. Beauchemin, "Technical note: Validation of a system for monitoring feeding behavior of dairy cows," *J. Dairy Sci.*, vol. 86, no. 11, pp. 3571–3574, 2003.

[7] H. Ardö, O. Guzhva, and M. Nilsson, "A CNN-based cow interaction watchdog," in *Proc. 23rd Int. Conf. Pattern Recognit.*, 2016, pp. 1–4.

[8] C.-H. Chen and K.-H. Liu, "Stingray detection of aerial images with region-based convolution neural network," in *Proc. IEEE Int. Conf. Consum. Electron.*, Jun. 2017, pp. 175–176.

[9] L. Fu, Y. Feng, Y. Majeed, X. Zhang, J. Zhang, M. Karkee, Q. Zhang, "Kiwifruit detection in field images using Faster R-CNN with ZFNet," *IFAC-PapersOnLine*, vol. 51, no. 17, pp. 45–50, 2018.

[10] C. Fraley and A. E. Raftery, "How many clusters? Which clustering method? Answers via model-based cluster analysis," *Comput. J.*, vol. 41, no. 8, pp. 578–588, 1998.

[11] A. Goel, "A study of different partitioning clustering technique," *Int. J. Sci. Res. Develop.*, vol. 2, no. 8, pp. 128–130, 2014.

[12] R. Chauhan, P. Batra, and S. Chaudhary, "A survey of density based clustering algorithms," *Int. J. Comput. Sci. Technol.*, vol. 5, no. 2, pp. 169–171, 2014.

[13] G. Münz, S. Li, and G. Carle, "Traffic anomaly detection using K-means clustering," in *Proc. Perform., Rel. Dependability Eval. Commun. Netw. Distrib. Syst., 4GI/ITG Workshop MMBnet*, Hamburg, Germany, Sep. 2007, pp. 1–8.

[14] L. Eriksson, T. Byrne, E. Johansson, J. Trygg, and C. Vikström, *Multi- and Megavariate Data Analysis: Basic Principles and Applications*, 3rd ed. Malmö, Sweden: MKS Umetrics AB, 2013.

[15] N. Suresh and K. Arulanandam, "An efficient K-means clustering algorithm using euclidean distance techniques," *Int. J. Adv. Res. Comput. Commun. Eng.*, vol. 6, no. 11, pp. 259–266, 2017.

[16] A. Singh, A. Yadav, and A. Rana, "K-means with three different distance metrics," *Int. J. Comput. Appl.*, vol. 67, no. 10, pp. 13–17, 2013.

[17] J. MacQueen, "Some methods for classification and analysis of multivariate observations," in *Proc. 5th Berkeley Symp. Math. Statist. Probab.*, vol. 1, 1967, pp. 281–297.

[18] K. Wagstaff, C. Cardie, S. Rogers, and S. Schröedl, "Constrained k-means clustering with background knowledge," in *Proc. Int. Conf. Mach. Learn.*, vol. 1, 2001, pp. 577–584.

[19] D. Birant and A. Kut, "ST-DBSCAN: An algorithm for clustering spatial-temporal data," *Data Knowl. Eng.*, vol. 60, no. 1, pp. 208–221, 2007.

[20] B. Borah and D. K. Bhattacharyya, "An improved sampling-based DBSCAN for large spatial databases," in *Proc. Int. Conf. Intell. Sens. Inf. Process.*, Chennai, India, 2004, pp. 92–96.

ZOOL HILMI ISMAIL received the B.Eng. and M.Eng. degrees in mechatronics engineering from Universiti Teknologi Malaysia, Skudai, Malaysia, in 2005 and 2007, respectively, and the Ph.D. degree in electrical engineering from Heriot-Watt University, Edinburgh, U.K., in 2011. He was appointed as a Senior Lecturer with Universiti Teknologi Malaysia, in 2011. He is currently a Research Member with the Malaysia-Japan International Institute of Technology and the Center for Artificial Intelligence and Robotics, UTM Kuala Lumpur. He was also appointed as a Visiting Researcher at Kyoto University, in 2014, and at the Jordan University of Science and Technology, in 2016. His current research interests are in the areas of edge computing, model-predictive control, path-planning, and task allocation based on deep-reinforcement learning. He received the number of awards from International RoboCup Competitions (Service Robot Category). He is a Registered Professional Engineer under the Board of Engineers Malaysia, a member of Society for Underwater Technology, The Institution of Engineering and Technology, the Institute of Electrical and Electronics Engineers—Oceanic Engineering Society, Asian Control Association, and a member and registered Chartered Marine Engineer of the Institute of Marine Engineering, Science and Technology. In 2017, he was appointed as one of committee member for International RoboCup@Home Education.

ALAN KH'NG KEAN CHUN received the B.Eng. degree in electronic systems engineering from the Malaysia-Japan International Institute of Technology, Universiti Teknologi Malaysia, in 2019. He is currently with the Circuit Design Team, Intel Company stationed at Penang, Malaysia. Previously, he was a Research Student in artificial intelligence and robotics under the supervision of MJIIT. His interest includes model-based clustering method in neural networks for various objects with dynamic and static behavior.

MOHD IBRAHIM SHAPIAI RAZAK received the M.Eng. degree from the University of York, U.K., in 2007, and the Ph.D. degree in machine learning from Universiti Teknologi Malaysia, in 2013. From March 2010 to April 2010, he was a Visiting Researcher at the Graduate School of Information, Production and Systems, Waseda University, Japan, under the supervision of Prof. Dr. J. Watada. From June 2012 to July 2012, he was with the Faculty of Engineering, Leeds University, U.K., under the supervision of Prof. Dr. V. Toropov. He is currently a Senior Lecturer with Universiti Teknologi Malaysia, and also a Researcher with the Center of Artificial Intelligence and Robotics (CAIRO). His research interests are artificial intelligence, machine learning, brain–computer interface, and swarm intelligence. He has also been appointed as a Certified NVIDIA Deep Learning Instructor.