From Nature to UAV - A Study on Collision Avoidance in Bee Congregation

Nahin Hossain Uday[a], MD Zahid Hasan[b], Rejwan Ahmed[c], DR. Debajyoti Karmaker[d]

[a] *Department of Computer Science & Engineering, American International University-Bangladesh, Dhaka, Bangladesh*
[b] *Department of Computer Science & Engineering, American International University-Bangladesh, Dhaka, Bangladesh*
[c] *Department of Computer Science & Software Engineering, American International University-Bangladesh, Dhaka, Bangladesh*
[d] *Faculty of Science & Technology, American International University-Bangladesh, Dhaka, Bangladesh*

Abstract

Swarm Intelligence (SI) is the new powerful era for the revolution of Artificial Intelligence of modern science. SI is the collective information and analysis of the behavior of biological existence like honey bee, bat, cat, ant, fireflies etc. The conception of SI is the collective behavioral information of living entities with a very limited set of rules (Bharne and Prabhune, 2017). Nowadays, SI is used in different type of applications like stock market price movement, routing algorithm, treatment for breast cancer, cloud computing etc. In this paper, we have proposed a combination of Particle Swarm Optimization (PSO) algorithm, Convolutional Neural Network (CNN) algorithm and Artificial Bee Colony (ABC) Algorithm could deliver a method for avoid-

*Corresponding author.
Email addresses: nahinu95@gmail.com (Nahin Hossain Uday), jahid1234.jh@gmail.com (MD Zahid Hasan), rejwan.ahmed@live.com (Rejwan Ahmed), d.karmaker@aiub.edu (DR. Debajyoti Karmaker)

ing collision in Unmanned Aerial Vehicle (UAV). For analyzing the behavior of any living entity, we selected honeybees (Apis Mellifera) and worked on a small portion of it. We have captured the video of honeybees flying close to a hive (human-made artificial hive) while the entrance was temporarily sealed which resulted the "bee cloud". An exploration of the flight trajectories executed and a 3D view of the "bee cloud" constructed. We analyzed the behaviors of honeybees, specially on their speed and distance. The results showed that the loitering honeybees performed turns that are fully coordinated, and free of side-slips so thus they made no collision between themselves which inspired us to propose a method for avoiding collision in unmanned aerial vehicle. This paper gives the collective behavioral information and analysis report of the small portion of data set (honeybees), other methods of SI used, performance evaluation using the dataset, its advantages and future works to be needed.
Keywords: Artificial Bee Colony, Collision Avoidance, Convolutional Neural Network, Honeybee, Particle Swarm Optimization, Swarm Intelligence, Unmanned Aerial Vehicle.

1. Introduction

From our daily experience we can figure out that speedy driving around a corner can be a reason for collision. Sometime speedy driving causes the failure of maintaining a safe distance between the vehicles which ends with an accident due to collision. So maintaining speed and a safe distance are so important for avoiding collision in any kind of vehicle. But maintaining these two constraints often become tough. For solving these problems we can use the concept of swarm intelligence (SI) along with Convolutional Neural Network (CNN). SI belongs to bio inspired computing field. The SI algorithms are derived from the behaviour of birds, ants, firefly, fishes etc.

There are many algorithms are developed in this field along with huge number of applications. For example, Mahadeeswara and Srinivasan (2018) analyzed the turning flight characteristics of loitering honeybees (Apis Mellifera) in a semi-outdoor environment with a amount of bees flying in close to each other while the results revealed that loitering bees performed turns that are fully coordinated, and free of side slips, so no collision happened between those honeybees in their experiment.

Swarm Intelligence is being used in medical science also. For example, Dheeba et al. (2014) described about breast cancer where they added that sometime breast cancer detection experiments had resulted false positive results when mammogram finds something that looks like cancer but really its not cancer, even very qualified radiologists made it hard to interpret. To overcome this situation, Dheeba et al. (2014) introduced a novel intelligent classifiers that use texture information as input to classify the normal and abnormal tissues in mammograms using Particle Swarm Optimized Wavelet Neural Network. Their proposed Computer-aided detection and diagnosis (CAD) system was based on a pattern recognition system which could detect the abnormal regions very intelligently (Dheeba et al., 2014).

Many algorithms have been derived from SI, among them Particle Swarm Optimization (PSO) is one of the reputed population based algorithm derived from SI. PSO is mainly used for grouping or clustering objects. Su et al. (2009) proposed a novel algorithm of data projection named Swarm-

Inspired Projection (SIP), inspired from the foraging behaviors of doves and allowed to visually calculate the number of clusters existing in a data set. SIP allows each data pattern in a data set as a piece of crumb to be tossed to a swarm of doves on the ground while the doves would adjust their placement to compete for crumbs and slowly the swarm of doves would be divided into several groups due to the crumbs distributions which allowed to view the sprinkle plot of the final placements of the doves from where the authors could estimate the number of clusters placed in the data set (Su et al., 2009). Wang et al. (2019) introduced a novel particle swarm optimization (PSO) named cPSO-CNN for optimizing the hyper-parameter configuration of architecture-determined Convolutional Neural Network (CNN) by using a confidence function to improve the conventional PSO algorithms exploration capability. cPSO-CNN can redefine the scalar acceleration coefficients of PSO as vectors to improve adaptability for the variant ranges of CNN hyper-parameters with less computational cost (Wang et al., 2019).

Artificial Bee Colony (ABC) is another swarm intelligence algorithm inspired from the smart behavior of honeybees. In this working model, all honeybees are made into three groups- I. Employee Bees (EB), II. On-lookers Bees (OB), and III. Scouts(S), according to their activities but among them there is only one EB is responsible for one food source (Bharne and Prabhune, 2017). In ABC algorithm, the location of a food source illustrates a possible solution to the optimization problem and the nectar amount of a food source fits to the quality (fitness) of the solution (Karaboga and Akay, 2009). Mustaffa et al. (2014) utilized Artificial Bee Colony to optimize the parameters of least squares support vector machines.

PSO is a very powerful algorithm for clustering objects and ABC is responsible for finding the best solution, but for clustering the objects, first image processing or image classification needs to be done. The Convolutional Neural Network (CNN) is a kind of artificial neural networks (ANNs) and has upright performances in many tasks such as image classification, real-time object detection (Pham et al., 2020), and object segmentation (Bi et al., 2010). The impression behind the higher performance of CNN architectures is by means of the ability of convolutional layers for identifying local combinations of features, and assembling layers for assimilation of semantically similar features into one (Strumberger et al., 2019).

In this paper, we proposed a method is a combination of CNN, PSO and ABC algorithms to avoid collision in UAV which was mainly inspired from the behaviour of loitering honeybees. The rest of this paper is documented as follows: Section 2 presents our proposed method and architecture, subjects (motivations), experimental configuration, recording, and reconstruction of 3D trajectories, Section 3 presents detail analysis on honeybees speed and distance and our findings from this analysis, Section 4 presents discussion, and Section 5 presents conclusion and future work.

2. Materials & Methods

This section presents our proposed method which will provide a solution to collision avoidance in UAV. In our study, we proposed that combination of convolutional neural network (CNN), particle swarm optimization (PSO) and Artificial bee colony (ABC) algorithms could be implemented in UAV to support the constraints and avoiding the collision. These three algorithms are population-based algorithm. We mainly focused on the area where UAVs are prone to collide with each other. In our proposed method, CNN will take the image of the area as its input. CNN will work on UAV control by processing the image, calculating the safe distance, and operating the UAVs. CNN will generate the processed image which will be considered as input of PSO. PSO will be responsible for road identification through clustering identification, finding cluster wise best result and finding global best result. If CNN's safe distance calculation becomes vulnerable or risky, then Artificial Bee Colony (ABC) algorithm will work to find the best position for the UAVs which derives to collision avoidance. This whole process is shown in Figure 1.

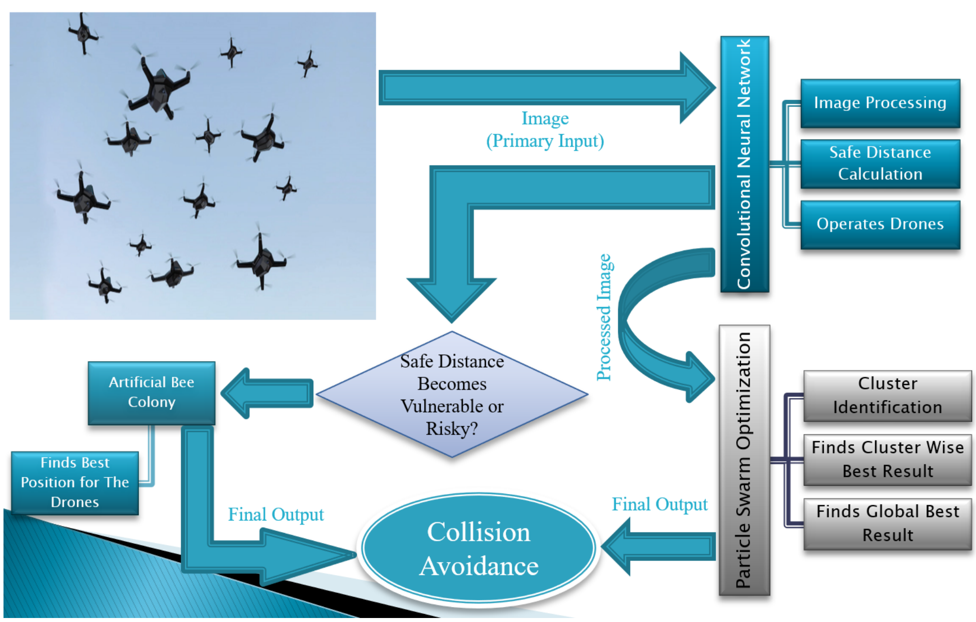


Figure 1: Collision avoidance using CNN, PSO & ABC algorithm

2.1. Subjects

For analyzing the behavior of any living entity, we selected honeybees and worked on a small portion of it. There are several motivations for selecting honeybees as our experimental subject. For example, bees never collide with each other while flying in close proximity, million years of evolution have made them successful in achieving flight behaviour and the significant part is they are very intellectual while their brains are less than two cubic millimetres in volume which is only about 0.0002 percent of our human brain. For experimenting the honeybee's behavior we visited a farm where the honey farmers collect the honey by providing human made hives (artificial hives) to their honeybees (Apis Mellifera).

2.2. Experimental Configuration

We temporarily blocked the entrance of a hive for experimenting their behavior like how they would communicate with each other without making any collision. By blocking their entrance the returning bees were temporarily denied to have entry into the hive but they were flying and loitering near to the entrance of the hive and attempting to make the entry into the hive which caused the 'bee cloud' as shown in the Figure 2.

Figure 2: Bee cloud view from a frame

Meanwhile we made the calibration using a checker board (as shown in the Figure 3) for two cameras from a different angle and distance.

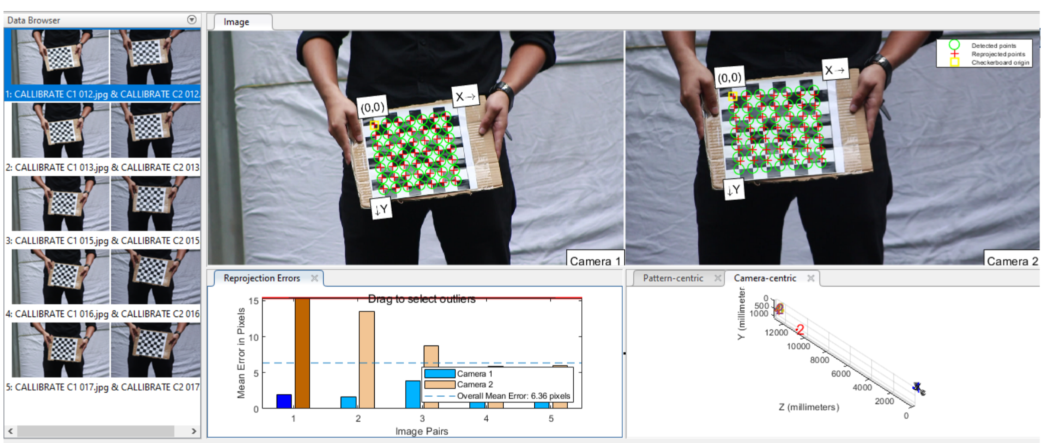


Figure 3: Calibration using a checker board

The measurement of a single rectangle of this checker board was 1 inch or 26.4mm, the distance between two cameras was 8cm and the heights from the ground of the both cameras were 30cm. We filmed that bee cloud by using two synchronized digital cameras (canon) to capture the stereo data as shown in animated Figure 4. The cameras recorded video at 29.970 fps with 1920X1080 (Width X Height) pixels resolution.

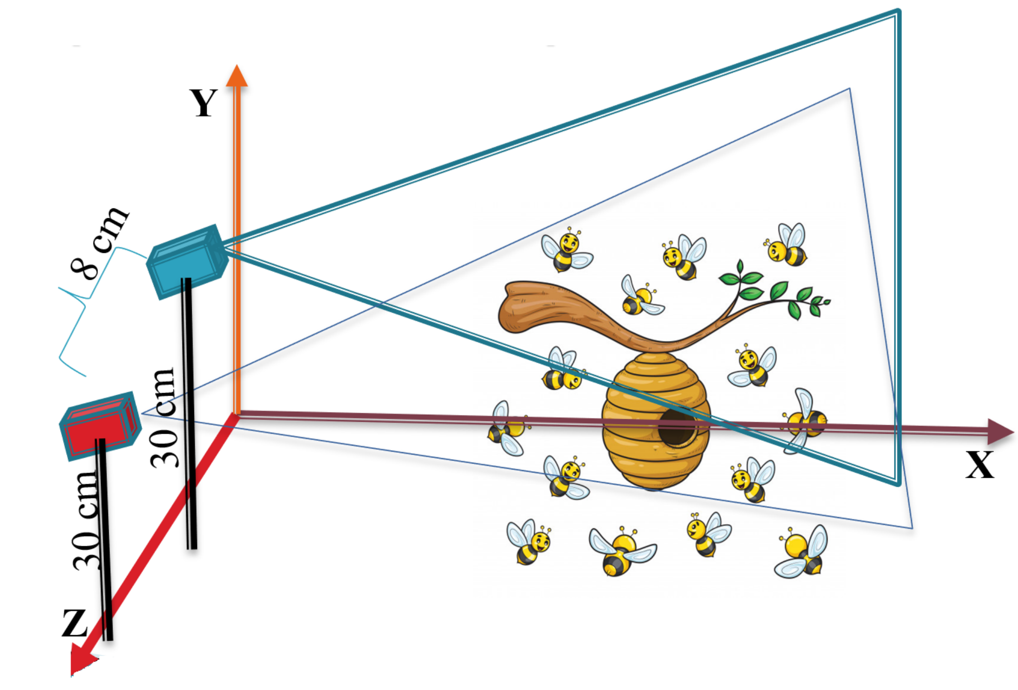


Figure 4: Experimental setup

2.3. Recording

After performing the stereo camera calibration we filmed the video by those two cameras with a good synchronization so that we could clearly identify each bee's head and tail positions graphically in each frame to acquire the bee's position coordinates in each view. First, we split the video of both cameras into several durations. Then we took same split of the video of both cameras and converted each video into 696 number of frames. Then we selected same 15 (approximately 0.5 second) suitable frames from each camera's generated frames for our experiments. Each frame contained 0.033 second. In Figure 5, there is a video of those 15 frames where the honeybees trajectory paths are showed.

Figure 5: Bees trajectory (video)

2.4. Reconstruction of 3D Trajectories

We analyzed the bees position using “Labelme” by manually annotating each bee’s position from frame to frame by drawing a rectangle on each bee with different id name (as shown in Figure 6) which resulted the JSON files consist of the coordinates of the loitering honeybees for each frame.

Figure 6: Bee annotation

We wrote a program in Python which took the JSON file as input and read those JSON files and play the frames sequentially so that it should be appeared like a video where every bee’s position was traced and clearly identified. And later we reconstructed a 3D view of the ”bee cloud” of 101 honeybees covered in the frames$(1 - 15)$ as shown in the Figure 7.

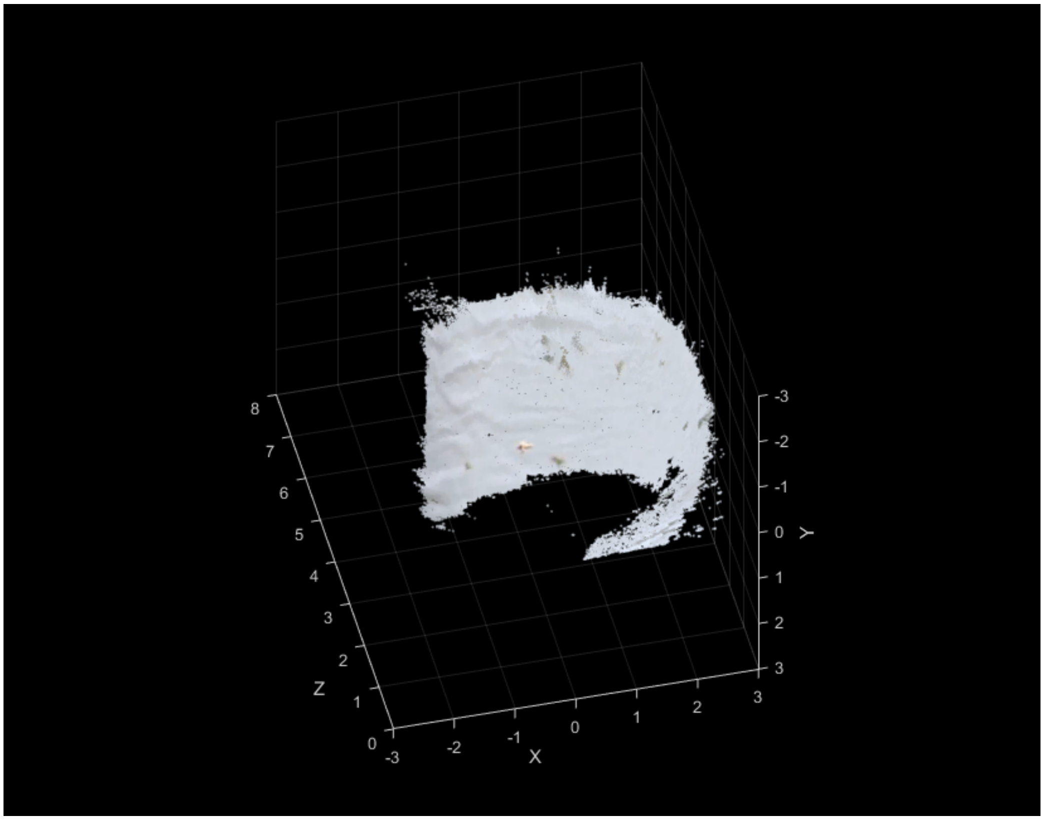


Figure 7: 3D view of the "Bee Cloud" of 101 honeybees

To compute linear distance between honeybees we used the Euclidean distance formula from (Anton and Rorres, 2014) is shown in Equation (1)

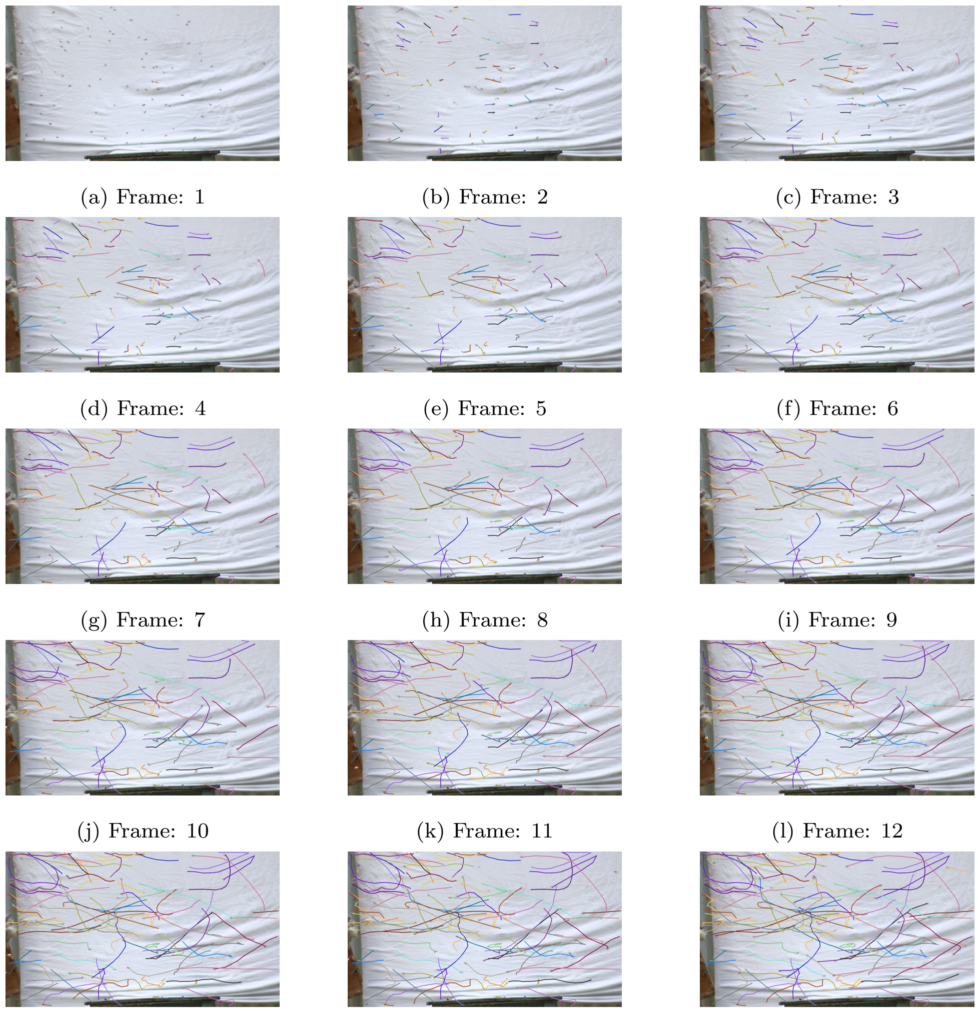
$$d(p,q) = d(q,p) = \sqrt{\sum_{i=1}^{n}(q_i - p_i)^2} \tag{1}$$

Here, p & q are two different points. We used distance(d) and time(t) to compute the speed(s) of the honeybees is shown in Equation (2)

$$s = d/t \tag{2}$$

3. Results

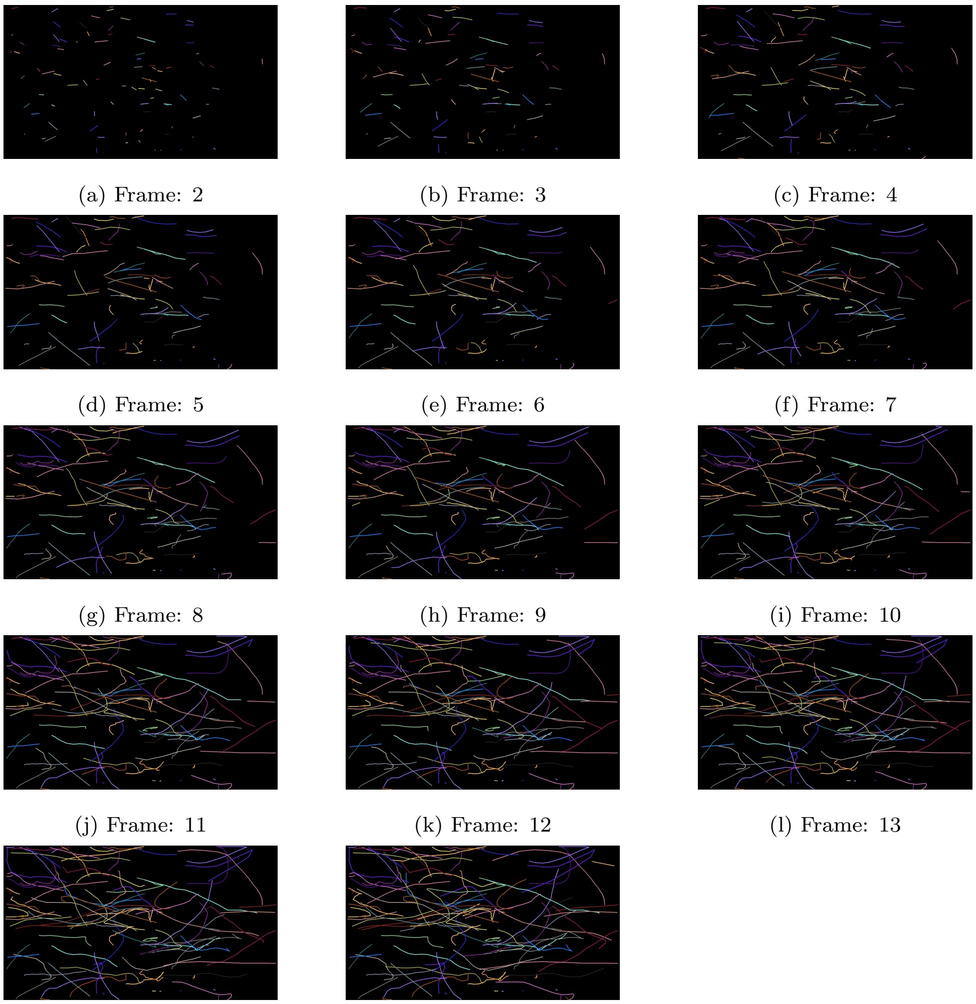
We plotted the trajectory of the bees (as shown in Figure 8) by tracing the midpoint which we saved in the JSON file. The bee trajectory frames were placed sequentially to make the trajectory video. As we took 15 frames so each frame contains 0.033 sec. In this fraction of second, we analysis the bee trajectory, its speed and the distance. We set up some criteria to identify these parameters.



(a) Frame: 1 (b) Frame: 2 (c) Frame: 3
(d) Frame: 4 (e) Frame: 5 (f) Frame: 6
(g) Frame: 7 (h) Frame: 8 (i) Frame: 9
(j) Frame: 10 (k) Frame: 11 (l) Frame: 12
(m) Frame: 13 (n) Frame: 14 (o) Frame: 15

Figure 8: Bees Trajectory

In Figure 8, The bees trajectories are directly shown from the actual frame to frame. Then we change the background to have a clear vision of the bee trajectories as shown in the Figure 9.In Figure 9, we didn't count $Frame$: 1 as it is the start of trajectory so it reflects nothing in this clear vision background. It looks meaningless at a first glance but when we get the data as per our parameters, we found a lot of facts related to this trajectory paths.



(a) Frame: 2 (b) Frame: 3 (c) Frame: 4

(d) Frame: 5 (e) Frame: 6 (f) Frame: 7

(g) Frame: 8 (h) Frame: 9 (i) Frame: 10

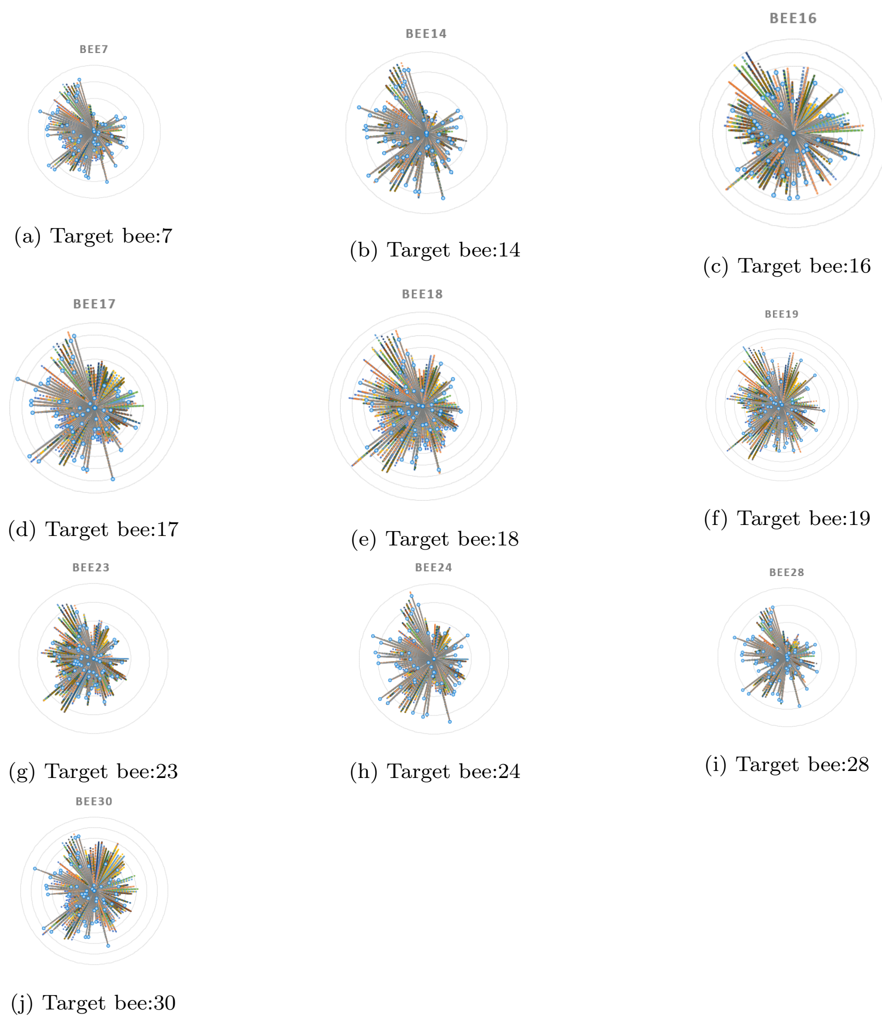
(j) Frame: 11 (k) Frame: 12 (l) Frame: 13

(m) Frame: 14 (n) Frame: 15

Figure 9: Bees Trajectory in clear Vision

First, we generated 3D positions of the bees in our frames. To do that we performed camera calibration. We used Matlab (2019) for camera calibration. There are some steps in 3D position generation. First of all, we generate rectified camera frames. From the rectified camera frames we generate disparity map. Disparity map generation is important for merging the two camera views. Then we generated 3D visualization and positions of the bees. The area is plotted in millimeter scale. The axis information then converted to a JSON file and later it converts to CSV file format.

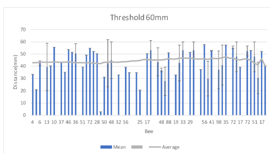
After that, we randomly selected 10 bees and calculated distance for every bee from our selected bees. The selected bees were random but there was a criterion that the selected bees are present in every frame. We found many bees which were present in every frame in these 15 frame season. We call our selected bees "Target bees". From target bees we calculated distance for every other bees by using Euclidean distance theorem from Equation (1). We plotted this in a radar view graph for every frame as shown in the Figure 10.



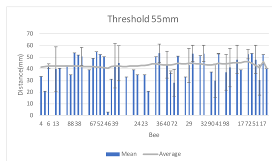
(a) Target bee:7 (b) Target bee:14 (c) Target bee:16 (d) Target bee:17 (e) Target bee:18 (f) Target bee:19 (g) Target bee:23 (h) Target bee:24 (i) Target bee:28 (j) Target bee:30

Figure 10: 3D Radar view graph for different target bees

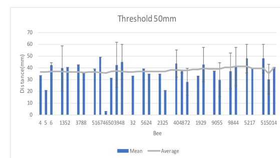
After this step we generated CSV files for further analysis. We implemented KNN for our target bees. We considered a threshold of different radius. A threshold is a radius of surrounding area of our targeted bees. We considered every bee entering this threshold. The entering bees were the nearest neighbor for our target bees as we implemented KNN algorithm. We calculated their speed and distance from our target bees. Graphs in Figure 11 are the bees distance bar chart. We reduced the threshold value and checked the results for different thresholds and we got the result that bees were maintaining this minimum distance from other bees. Sometimes the distance was crossed but the majority number of bees were maintaining a specific distance.



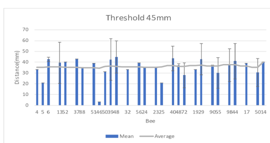
(a) Threshold 60mm



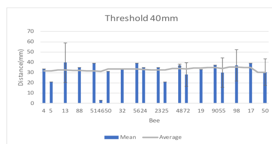
(b) Threshold 55mm



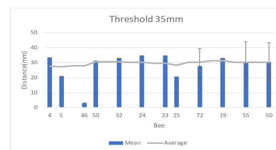
(c) Threshold 50mm



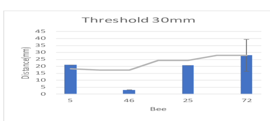
(d) Threshold 45mm



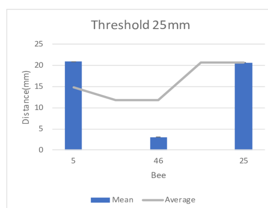
(e) Threshold 40mm



(f) Threshold 35mm



(g) Threshold 30mm



(h) Threshold 25mm

Figure 11: Distance comparison for different threshold according to target bees

Here the average distance is plotted as a line over the bar charts to understand how much distance is maintained by individual bees. The error bars are standard deviation for the distance. It is clear by this charts that bees are maintaining a distance between 30mm to 35mm, this threshold values gives us the best result. We found majority of the bees were maintaining a minimum distance under this threshold values. For clear representation of the error bars of these graphs of distance comparison for different threshold according to the target bees, we generated a graph (as shown in fig. 12) of error vs threshold where X axis defines the sequence of graphs from Figure 11 and Y axis defines the threshold values.

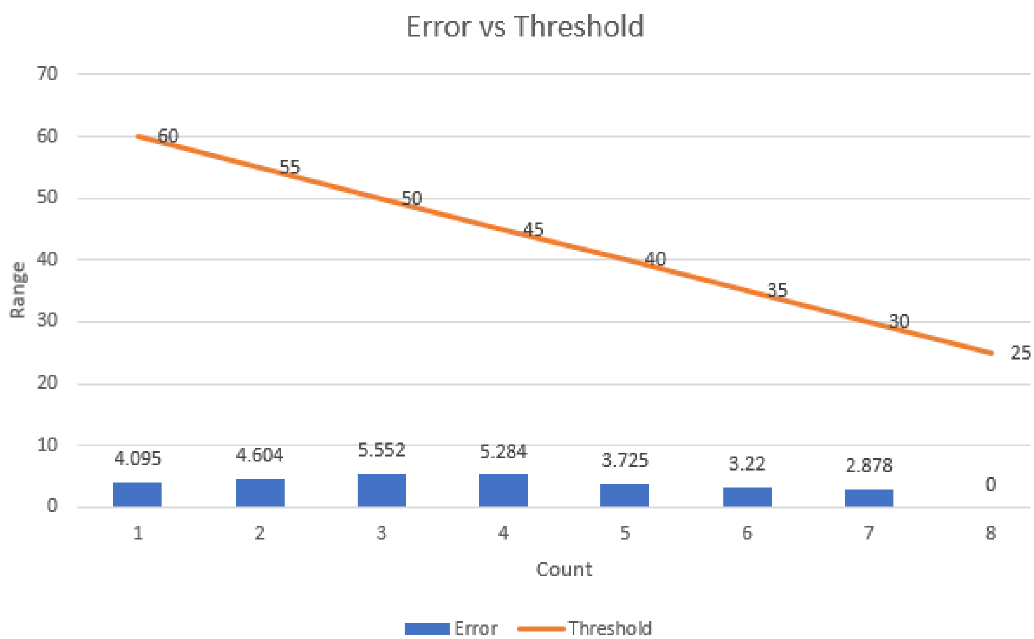
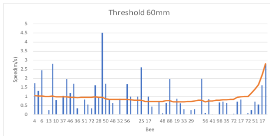
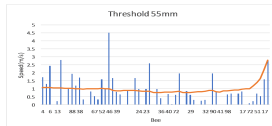


Figure 12: Error vs Threshold

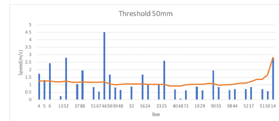
We also calculated the speed for every bee that came into the threshold area and generated bar chart to show the speed of the bees. Graphs in Figure 13 represent the speed comparison of the loitering honeybees for different threshold according to target bees.



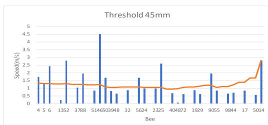
(a) Threshold 60mm



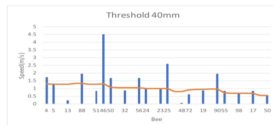
(b) Threshold 55mm



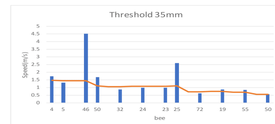
(c) Threshold 50mm



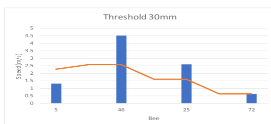
(d) Threshold 45mm



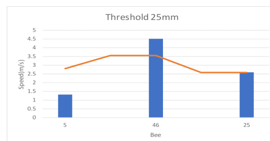
(e) Threshold 40mm



(f) Threshold 35mm



(g) Threshold 30mm

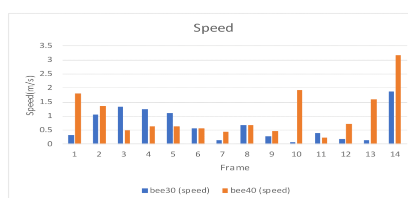


(h) Threshold 25mm

Figure 13: Speed comparison for different threshold according to target bees

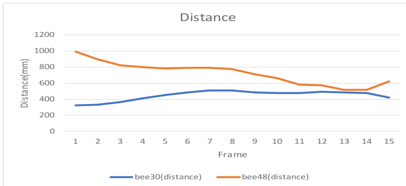
There is no variation in speed but it gives an interesting result we considered speed along with distance for respectively two bees. Our target bees and any other bees within the threshold are selected for this experiment. We considered speed along with their distance in our chosen15 frames. We considered speed and distance in every frame and their rate of change for these two bees. The following graphs in Figures 14 to 23 are representing the result of this experiment.

(a) Distance comparison on bee30 vs bee40

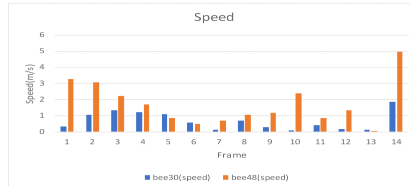


(b) Speed comparison on bee30 vs bee40

Figure 14: Speed & Distance comparison on bee30 vs bee40

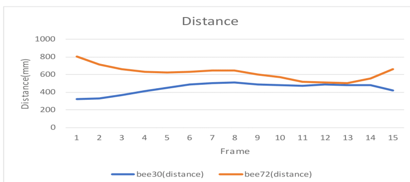


(a) Distance comparison on bee30 vs bee48

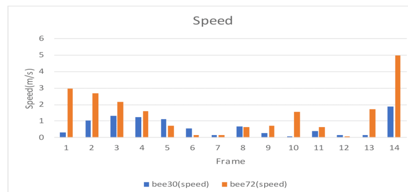


(b) Speed comparison on bee30 vs bee48

Figure 15: Speed & Distance comparison on bee30 vs bee48



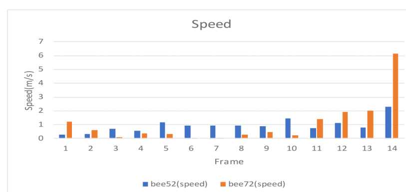
(a) Distance comparison on bee30 vs bee72



(b) Speed comparison on bee30 vs bee72

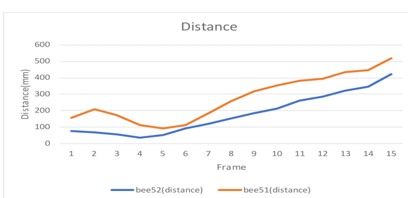
Figure 16: Speed & Distance comparison on bee30 vs bee72

(a) Distance comparison on bee52 vs bee72

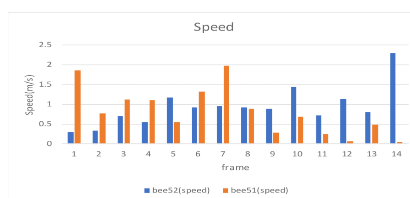


(b) Speed comparison on bee52 vs bee72

Figure 17: Speed & Distance comparison on bee52 vs bee72

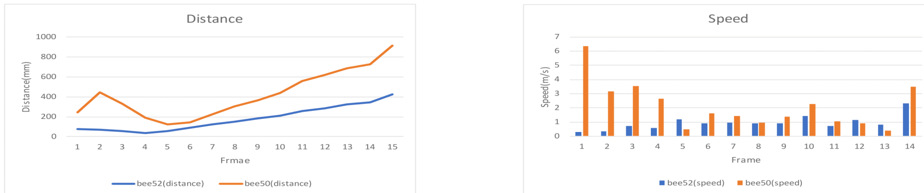


(a) Distance comparison on bee52 vs bee51



(b) Speed comparison on bee52 vs bee51

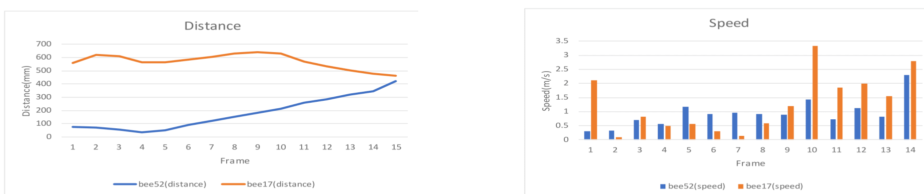
Figure 18: Speed & Distance comparison on bee52 vs bee51



(a) Distance comparison on bee52 vs bee50

(b) Speed comparison on bee52 vs bee50

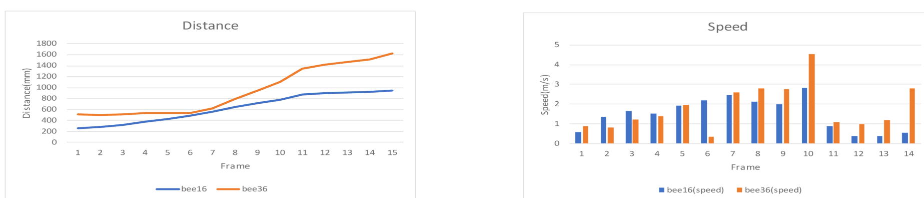
Figure 19: Speed & Distance comparison on bee52 vs bee50



(a) Distance comparison on bee52 vs bee17

(b) Speed comparison on bee52 vs bee17

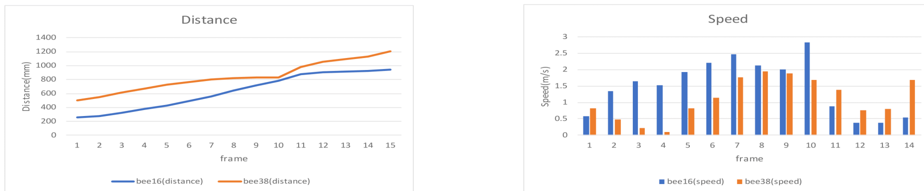
Figure 20: Speed & Distance comparison on bee52 vs bee17



(a) Distance comparison on bee16 vs bee36

(b) Speed comparison on bee16 vs bee36

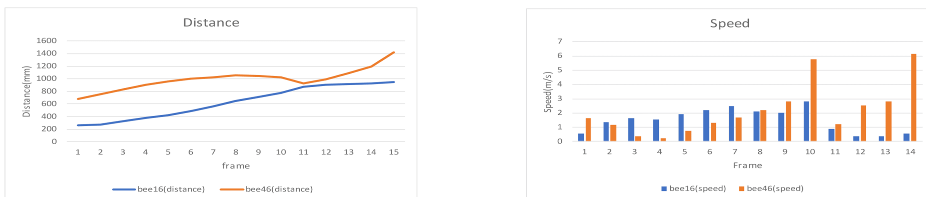
Figure 21: Speed & Distance comparison on bee16 vs bee36



(a) Distance comparison on bee16 vs bee38

(b) Speed comparison on bee16 vs bee38

Figure 22: Speed & Distance comparison on bee16 vs bee38



(a) Distance comparison on bee16 vs bee46

(b) Speed comparison on bee16 vs bee46

Figure 23: Speed & Distance comparison on bee16 vs bee46

From these graphs we can clearly see that, when two bees came closer, they reduce their speed dramatically. Most of the time the two bees maintain a slow speed. When the distance increases, the bees also increase their speed. In some frames we can see that one bee reduces speed dramatically but the other bee did not. But it was not often. Often both bees reduced speed when they came closer. This is for avoiding collision which is the main goal of our project. By analyzing these graphs, we can come into a conclusion that, the bees maintain a minimum distance with each other. We took the measure also which is explained previously. So, we can state that when two bees came closer, they reduced their speed and maintained a distance to avoid collision.

4. Discussion

For the experimental setup, we selected an open field with a covered area where bees were dancing. A white background proposed for the covered area. For making 3D video we used 2 cameras to generate 3D video. From ground to the target area, it was difficult to do 3D video by keeping the measurement correct for both cameras. The camera position and distance from the ground to camera height was challenging. The focus for both of the camera was correct to generate 3D video. Bees entry into the hive was blocked for temporarily so that bee could fly in the target zone. As cameras were set up according to the target zone, the flying position of the bee was recorded for a while. Several videos were taken in time difference. In that mean time the camera position and frame rate were same. After analyzing the videos, we selected the best video where the number of bees were better in the target area. Bee were flying so some of the bee were out of the frame in sometimes and again included after some time. Maximum bees were flying in the target frame.

We selected the same frame from both of the cameras where the bees were in same position for cameras. By calculating their 3D position, we generated some speed graph, distance graph individually and combined the graph to determine the process more visualize, assumed some threshold radius to show their movement processing, how the interact one another to avoid collision. As the path of the bee was nonlinear so we can avoid the collision by their nature. If we follow the linear position of the bee it would create a problem for collision avoidance in real life. Because UAV are not moving in a straight direction and if they move in a straight direction the speed and turning speed will be same for UAV which is not a real-life solution.

A definite amount of distance and limit of speed, helped the bees to avoid collision. A certain amount of area and distance they maintained. The speed was decreased by their position to a certain time. In maximum time both of the bee's compromised to reduce their speed.

5. Conclusion

From exploring the result, it can be said that bee maintains a safe distance to avoid collision and it was 35mm. It was called a threshold value to determine the safe distance. As collision avoidance occurs so another term called speed will must include to calculate the result. It was almost 1 m/s to maintain their speed. After they accelerate and decelerate their speed range, they find a solution to avoid collision. It can be called game of cooperative. It means two bees reach a point to compromise their decision. We use convolutional neural network for image classification and particle swarm optimization for clustering and ABC for finding best position. Convolutional neural network implemented by C4M3F2 to detect actual object. Particle swarm optimization used to make a cluster by the algorithm of KNN. A threshold value generated to determine the minimum distance. For finding best position a formula given with the help of ABC algorithm. By proposing three methods with our experimental result, it will generate a model to avoid collision for UAV. Our result section shows that real visualization of bee nature and the mechanism for collision avoidance and the graphs represent the actual measurement of our experiment. It can clearly visualize the speed, distance and threshold combination that help the mathematical accuracy.

For developing the proposed method in future, we have selected some area for improvement. The acceleration of bee will help more to reduce collision in future. Magnitude of centrifugal force and center of curvature will help to develop the speed. Micro-controller can be used for more controlled in some density area. These were some future development area of our paper.

References

Howard Anton and Chris Rorres. *Elementary Linear Algebra: Applications Version*. Wiley, eleventh edition, 2014. ISBN 9781118434413 1118434412 9781118474228 1118474228.

P. K. Bharne and S. S. Prabhune. Survey on combined swarm intelligence and ann for optimized daily stock market price. In *2017 International Conference on Soft Computing and its Engineering Applications (icSoftComp)*, pages 1–6, Dec 2017. doi: 10.1109/ICSOFTCOMP.2017.8280083.

Q. Bi, H. Zhu, and T. Yue. Application of chaos ant colony neural network in slope stability analysis. In *2010 International Conference on Computational and Information Sciences*, pages 645–648, Dec 2010. doi: 10.1109/ICCIS.2010.162.

J. Dheeba, N. Albert Singh, and S. Tamil Selvi. Computer-aided detection of breast cancer on mammograms: A swarm intelligence optimized wavelet neural network approach. *Journal of Biomedical Informatics*, 49: 45 – 52, 2014. ISSN 1532-0464. doi: https://doi.org/10.1016/j.jbi.2014.01.010. URL http://www.sciencedirect.com/science/article/pii/S1532046414000124.

Dervis Karaboga and Bahriye Akay. A comparative study of artificial bee colony algorithm. *Applied Mathematics and Computation*, 214(1):108 – 132, 2009. ISSN 0096-3003. doi: https://doi.org/10.1016/j.amc.2009.03.090. URL http://www.sciencedirect.com/science/article/pii/S0096300309002860.

Mandiyam Y. Mahadeeswara and Mandyam V. Srinivasan. Coordinated turning behaviour of loitering honeybees (apis mellifera). *bioRxiv*, 2018. doi: 10.1101/332379. URL https://www.biorxiv.org/content/early/2018/06/04/332379.

Zuriani Mustaffa, Yuhanis Yusof, and Siti Sakira Kamaruddin. Enhanced artificial bee colony for training least squares support vector machines in commodity price forecasting. *Journal of Computational Science*, 5(2):196 – 205, 2014. ISSN 1877-7503. doi: https://doi.org/10.1016/j.jocs.2013.11.004. URL http://www.sciencedirect.com/science/article/pii/S1877750313001300. Empowering Science through Computing + BioInspired Computing.

V. D. Pham, Q. Nguyen, H. Nguyen, V. Pham, V. M. Vu, and Q. Bui. Convolutional neural network—optimized moth flame algorithm for shallow landslide susceptible analysis. *IEEE Access*, 8:32727–32736, 2020. ISSN 2169-3536. doi: 10.1109/ACCESS.2020.2973415.

I. Strumberger, E. Tuba, N. Bacanin, M. Zivkovic, M. Beko, and M. Tuba. Designing convolutional neural network architecture by the firefly algorithm. In *2019 International Young Engineers Forum (YEF-ECE)*, pages 59–65, May 2019. doi: 10.1109/YEF-ECE.2019.8740818.

Mu-Chun Su, Shi-Yong Su, and Yu-Xiang Zhao. A swarm-inspired projection algorithm. *Pattern Recognition*, 42(11):2764 – 2786, 2009. ISSN 0031-3203. doi: https://doi.org/10.1016/j.patcog.2009.03.020. URL `http://www.sciencedirect.com/science/article/pii/S003132030900123X`.

Yulong Wang, Haoxin Zhang, and Guangwei Zhang. cpso-cnn: An efficient pso-based algorithm for fine-tuning hyper-parameters of convolutional neural networks. *Swarm and Evolutionary Computation*, 49:114 – 123, 2019. ISSN 2210-6502. doi: https://doi.org/10.1016/j.swevo.2019.06.002. URL `http://www.sciencedirect.com/science/article/pii/S2210650218310083`.