Imperial College London

# Evaluating the ability of deep learning models to track markerless bumblebees

Paul Cabrisy

August 2022

A thesis submitted for the partial fulfillment of the requirements for the degree of Master of Science at Imperial College London

Submitted for the MSc in Computational Methods in Ecology and Evolution

## Declaration and Acknowledgement

I declare the work presented in this thesis to be my own. All videos and coding scripts are available on request and can be found on my github account at: https://github.com/paulcabrisy/CMEE_Project.

I would like to personally thank my supervisor Peter Graystock for his guidance, encouragements and support with this project. It has been immensely appreciated. I also acknowledge the computational resources and support provided by the Imperial College Research Computing Service (http://doi.org/10.14469/hpc/2232).

Paul Cabrisy, 25th August 2022.

## Lexicon

**ANN:** Artificial neural networks
**CPU:** Central processing unit
**DLC:** DeepLabCut
**GLM:** Generalized linear model
**GPU:** Graphics processing unit
**GUI:** Graphical user interface
**HPC:** High performance computing
**Networks:** Referring to the residual neural networks used in this study
**PLA:** Polylactic acid
**ResNet:** Residual neural network
**RMSE:** Residual mean squared error

## Keywords

Animal tracking, artificial neural networks, bumblebee, computer vision, DeepLabcut, deep learning, machine learning, markerless pose estimation.

## Abstract

Neural networks are widely used in behavioural studies to track multiple animals from videos and are valuable for tracking individuals within groups to help our understanding of social behaviour. However, animal tracking often relies on physical markers being put on body parts, which is not easily done on insects. With advancements in computer vision and new machine learning tools, there are now markerless ways of tracking animals which could be used to study social insect behaviour. This project aims to show that neural networks can be used for tracking fast moving insects like bumblebees in an open field type experiment. Tracking data from body part coordinates can be used to extract information from videos such as bumblebee counts within groups. In this study, training videos were collected using two camera setups to have multi-animal scenes with more or less occlusions. Three types of neural networks (ResNet50, ResNet101 and Dlcrnet_ms5) were compared after being trained across a range of iterations to optimise their training and find at what iteration their performance measured in pixel error was lowest. It was found that these neural networks differed in their ability to identify and track bumblebees. Larger numbers of training iterations were able to improve the performance of these networks and after comparing pixel errors from the training results, Dlcrnet_ms5 was found to be more accurate at identifying bumblebees. This network was then chosen to analyse novel videos and was better able to track bumblebees from recordings with less occlusion scenes.

## Introduction

Animal behaviour is determined by the interactions of individuals with others in relation to their environment and affects the organization of populations which have major implications on our ecosystems (Dell et al., 2014). There is a need to better understand interactions between individuals and patterns observed in social groups like eusocial insects, to gain insight into how animals respond to the current changing environment (Dell et al., 2014). However, environmental conditions, the size and speed of the animal or body part of interest and the spatial range all complicate the quantification of animal movement (Janisch et al., 2021). Therefore, there is a need to select the right tools to analyse animal behaviour to improve our understanding.

Video recordings can be used for studying animal behaviour and collecting data on fast moving animals like insects (Ratnayake, Dyer and Dorin, 2021). However, extracting information from videos is time consuming and prone to human error (Ratnayake, Dyer and Dorin, 2021). Moreover, analysing and quantifying insect behaviour by human observation can be challenging due to their small anatomy and has limitations such as being prone to individual biases and inconsistencies (Traner, Chandak and Raman, 2021). The ability to use computer algorithms to annotate behaviours would increase experimental reliability since the performance of machines vary less than that of humans and unlike humans machines can focus simultaneously on multiple individuals (Sclocoo et al., 2021).

Many machine learning algorithms use deep neural networks for annotating and analysing images within videos (Lou and Shi, 2020). A neural network is composed of a predetermined number of nodes that are processing units organised into layers (**figure 1A**). Layers are connected to each other with weights attributed to each node, a set of functions processes these weights to pass an output result to the next layer until reaching the final output layer (**figure 1B**). After reading the input data and doing calculations on it the network learns the patterns of the input data which is done repeatedly while updating input data with the new learned results (Nielsen, 2018). Epochs describe these loops from the input data to the output data going back to the input (Nielsen, 2018). Networks require several epochs to be trained with each epoch having multiple iterations that are the quantity of batches (training samples) required to complete one epoch (Nielsen, 2018). Computer vision has made significant advencements with neural networks being used in image classigication and recognition (Koushik, J., 2016). However, to solve more difficult problems, more layers are added to the architectures of networks which makes their training harder (Koushik, J., 2016). ResNets are deep learning networks which can address this issue of vanishing gradient occuring when a network has too many layers (He et al. 2016).

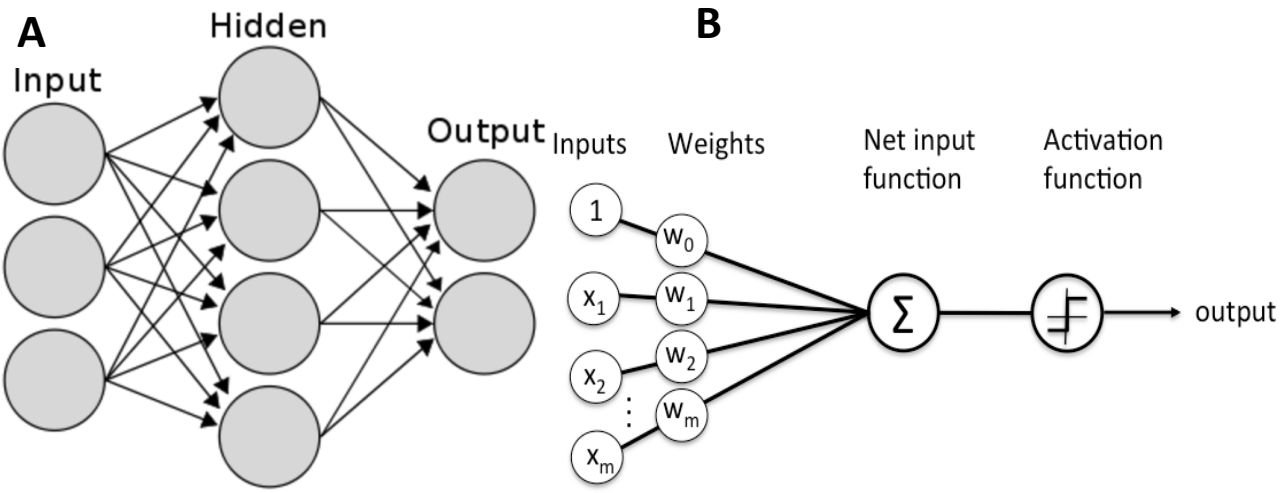



**Figure 1. A**) Simplified network example of three layers (input, hidden and output) with grey circles representing nodes. The number of nodes depends on the input data and network architect. **B**) Illustration showing weights attributed to the data in nodes of the input layer with functions processing the weights and passing them to the final output layer. Source: Baeldung (2020).

With the application of machine learning tools, neural networks have allowed markerless animal pose estimation algorithms to be developed (Arent et al., 2021 ; Lauer et al., 2022). Such algorithms are based on applications of artificial intelligence (Arent et al., 2021). The process of multi-animal pose estimation requires three steps: 1) localizing keypoints on individuals (pose estimation), 2) grouping the keypoints into distinct animals (assembly) and 3) tracking the movement of each animal (Lauer et al., 2022). Moreover, with the advances of high-resolution cameras and computer vision with graphical processing units (GPUs), progress has been made on analysing behaviours with greater speed (Sclocoo et al., 2021 ; Traner, Chandak and Raman, 2021).

The number of labelled frames in a training dataset and how effectively frames capture the behaviour of interest and the number of training iterations are major elements that will affect the tracking accuracy of a network (Clemensson et al., 2020). Therefore, the network performance is often increased in two ways either by improving the training dataset or by increasing the number of training iterations (Clemensson et al., 2020). Increasing training iterations is a common approach to improve inferences, but it comes with a trade-off between training time and performance increase (Labuguen et al., 2019). Knowing the number of iterations to use depends on the input data as smaller training datasets require higher iterations (Clemensson et al., 2020). So there is no fixed number for every study, but a good approach is to stop the training of networks when the training loss has plateaued (Labuguen et al., 2019). Like iterations, choosing the appropriate type of neural network to use for an experiment depends on the study conducted. Different networks can be designed and pretrained with the purpose of tracking specific behaviour or objects and there are now different networks available for tracking animals (Hardin and Schlupp, 2022).

When analysing videos, the complete or partial blockage of an animal, also called occlusion, is more frequent in multi-animal videos (Rodriguez et al., 2017 ; Lauer et al., 2022). When dealing with occlusions, one can only label visible body parts which requires labelling more images to increase the confidence of predictions made by the network (Mathis et al., 2018). Another approach is to guess where body parts (when occluded) are in the image but this affects the network performance by reducing the confidence in predictions made (Mathis et al., 2018). Therefore, social behaviours where occlusions are more common can be limitations that influence the inferences of networks. Recommendations on whether to teach networks to guess body parts by labelling occluded parts are specific to the behaviour studied (Mathis et al., 2018).

Here, I used a popular open source programme called DeepLabCut (DLC) for tracking bumblebees in an open field setup by collecting videos of their foraging activity at the entrance of their nest. The primary goal was to validate the reliability of DLC at detecting bumblebees within groups by comparing bumblebee counts recorded by human observation with counts made by DLC. Two camera setups were used to compare different neural networks trained on a range of iterations to optimise their performance and test their accuracy at identifying bumblebees. I tested the hypothesis that neural networks can be trained to track bumblebees from video recordings; and secondly that the tracking performance of these networks varies between network type and the number of training iterations. A last hypothesis is that neural networks are better able to track bumblebees from multi-animal scenes with less occlusions. To test this hypothesis, I collected recordings of bumblebees displaying flight and landing movements and recordings of bumblebees restricted to walk linearly through clear tubes.

## Methods

### 2.1 Software and hardware

DeepLabCut is a modified version of DeeperCut a pose estimation CNN and is an open source Python toolbox that allows users to track a set of predefined animal body parts with little pre-labelled data (Nath et al., 2019). DLC can detect body parts in visual dynamic environments with varying backgrounds (Nath et al., 2019). This tool now enables pose estimation in 3D spaces (Traner, Chandak and Raman, 2021). Moreover, DLC can train a network to label images at a human-level by performing frame-by-frame detection with minimal training images and analyse behaviours with occlusions (Nath et al., 2019). DLC applies a residual neural network (ResNet) architecture, which is a deep architecture of stacked convolutional networks with identity short cuts (He et al., 2016). The weights of ResNets used with DLC have been pretrained on ImageNet which is a large increasing dataset of millions of hand-annotated images available for machine learning

tasks (He et al., 2016). This allows DLC to achieve accurate results for image detection and classification tasks with little training data (He et al., 2016). DLC also provides a graphical user interface (GUI) to facilitate labelling (Lauer et al., 2022).

This experiment was conducted by submitting scripts to the Imperial College London HPC cluster and a laptop was used to label frames with the DLC GUI. GPUs could not be used because specific builds for TensorFlow could not be installed on the GPU nodes of the cluster. Therefore, all trainings and evaluations of networks were done using CPUs instead which typically take 10 times longer than with GPUs. The number of CPUs and other HPC resources used for running jobs are indicated in **table A1**.

## 2.2 Comparative camera angles setups

To validate the tracking performance of DLC, bumblebees were filmed entering and leaving their nest in two different ways: by recording workers at the entrance of the nest and recording workers walking through clear tubes. The first camera setup enabled filming bumblebees at their nest entrance where workers would fly across the camera frame and create occlusions by climbing on top of each other. The second setup was to film bumblebees in a restricted space where workers could not fly or walk over each other to have videos with fewer occlusions.

Two buff-tailed bumblebee (*Bombus terrestris*) colonies of around 50 workers were placed under red light to attach unfiltered pollen traps by inserting two 20cm clear tubes of 12.8mm diameter to the nest entrance holes. The traps were 3D printed using white PLA filament according to the Biobest Trap Body 3D design (Judd et al., 2020) and were simply to create a white frame around the nest entrance for cameras to better focus on the bumblebees. The trap design was slightly modified to remove some of the plastic around the holes to maximise the visibility of the nest entrance (**figure 2A**). Two tubes were connected to the traps with a white 3D printed board attached below to create a background to help the cameras focus on bumblebees walking through the tubes (**figure 2A**).

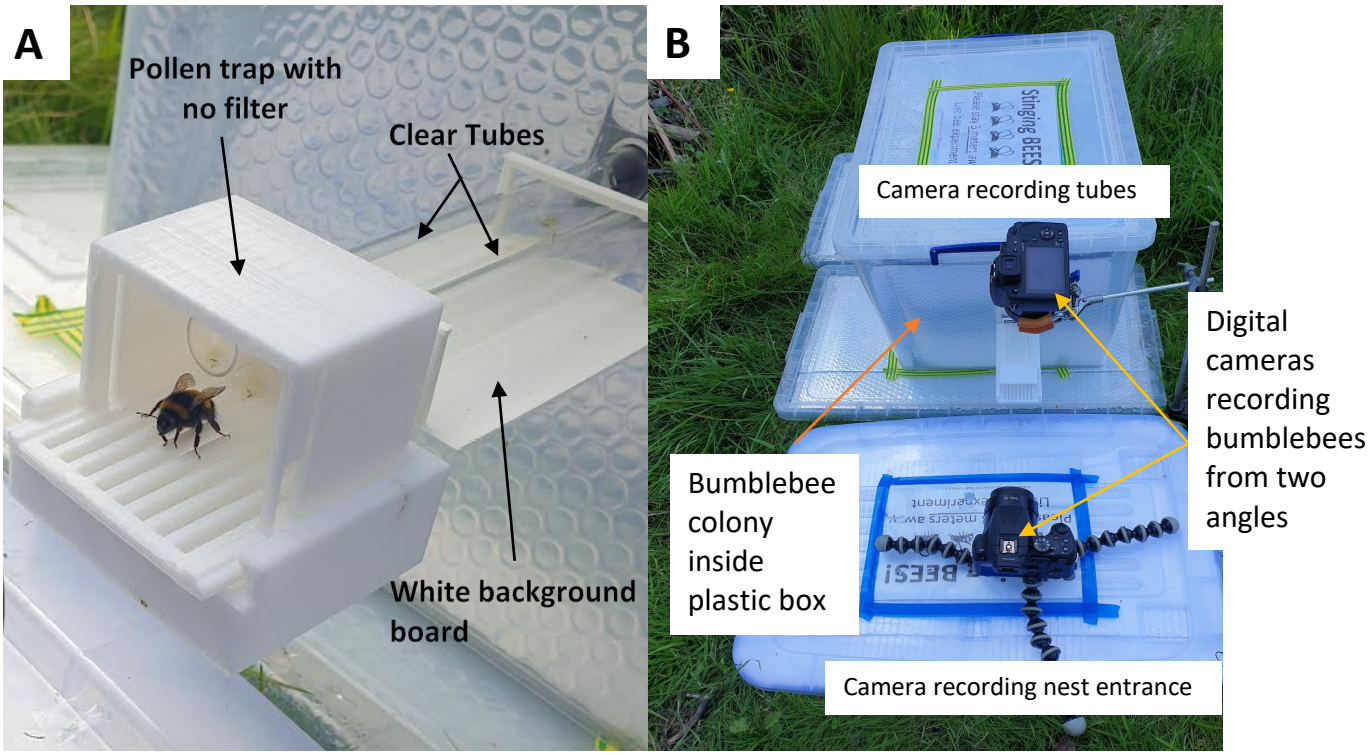



**Figure 2**. **A**) Photo of an unfiltered pollen trap connected to the nest entrance with two clear tubes attached to white background board. **B**) Photo of the two camera setups to film the behaviour of bumblebees from different camera angles.

The traps were closed with a trap closure to prevent the workers from escaping and pollen feeding was stopped 24 hours before deploying colonies outside to stimulate workers to collect new pollen. Each nest was placed inside a plastic box to protect it from precipitation and left in an open field on the Silwood Park campus in Ascot with adequate sun cover to reduce overheating (**figure 2B**). Once outside, the trap closure was removed to allow bumblebees to forage freely and orient themselves for 24 hours. The recordings were carried out for two weeks around midday with one to three recording sessions per day depending on weather conditions and using two digital cameras (Panasonic DC-FZ82) operated at a resolution of 3840 x 2160 pixels with a frame rate of 30 frames per second. To have minimal distortions in videos, recordings were done in a well-lit area and the resolution was maximised for these cameras to make future labelling easier.

## 2.3 Datasets, image extraction and annotation

The whole dataset consists of 42 ten minute videos of bumblebees from two colonies entering and leaving their nests. 12 videos were collected from one colony and used as training data and 30 were collected from the other colony to be used as validation datas to test the first hypothesis. Within the training videos, half of them were collected by placing a camera facing the entrance of the hive and the other six videos were recorded with the camera facing down at the clear tubes

(**figure 2B**). For the validation videos, 15 were collected with one camera setup and 15 more with the other. The training and validation datasets were from separate colonies so that videos from the one colony could be used to validate the training done on another and reduce behaviour biases.

When creating a DLC multi-animal project, a folder is created containing a configuration file with editable parameters for the training. Images were extracted from multiple videos at once using k-means clustering to extract frames displaying the most changes in the training videos and thereby sample frames with most varying behaviour. For each training video over 75,000 frames were extracted and 60 frames selected for labelling, giving 360 frames to label in total for the training videos of one camera setup. The frames were manually annotated using the DLC GUI (**figure 3**) with 6 body parts: head, thorax, abdomen, tail, right and left pollen baskets. These body parts were chosen for their ease to locate and identify on the bumblebees and the pollen baskets were labelled with the intention to collect foraging data, but due to time limitations this was not achieved.

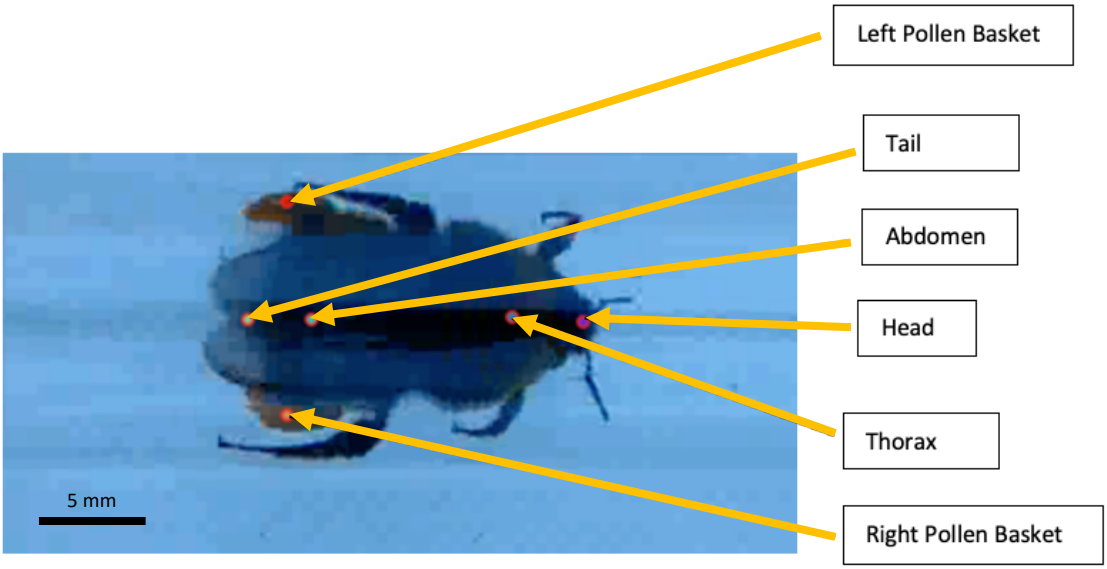



**Figure 3.** Bumblebee walking through a clear tube with six annotated body parts from the DeepLabCut GUI. The circled points represent the labelled key points that the neural networks are trained to track.

In the configuration file a maximum of 10 individuals were selected for labelling which is based on the highest number of workers observed in a single frame and the identity parameter was set to false so that bumblebees did not have a specific ID. Most body parts were labelled when clearly visible, however, some obstructed ones were estimated by eye in relation to other parts and bumblebee pose. Moreover, not all frames had annotations for all body parts and some had no annotations when workers were absent. In total, 75 annotations were added to the training videos of the nest entrance and 433 to the tube videos.

## 2.4 Training, evaluating and comparing neural networks

The default splitting ratio of data for training and validation was left unchanged with 95% training and 5% validation. Residual neural networks (ResNets) were selected for the training as these are expected to perform better than other networks when dealing with multiple individuals such as MobileNets or EfficientNets (AlDuwaile and Islam, 2021). Three ResNets (Dlcrnet_ms5, ResNet_50 and ResNet_10) were trained on the training videos across a range going from 5,000 to 20,000 iterations with 1,000 iteration intervals. This range was to see how the iteration parameter affects the performance of these networks with minimal training in order to compare them. The wall time on the CPU nodes was reached at 20,000 iterations, so the training could not be pushed further. Of note, 500 iterations is lowest iteration DLC allows to train networks but for the purpose of this study 5,000 was chosen as a minimal training baseline.

To compare their performances, these networks were trained on the same labelled frames and evaluated using mean Euclidean distances to compare their accuracy of labelled body parts (Nath et al., 2019). This is the residual mean squared error (RMSE) score given after evaluation which indicates how far a ground truth pixel labelled by the observer is from a predicted pixel (Nath et al., 2019). The lower the summed RMSE scores for an iteration the better the network was trained for predicting the coordinates of body parts. For each network, the summed RMSE scores of all annotations from the training videos were combined and compared between trainings on different iterations to see at what iteration the performance was greater. A polynomial model of degree four was then fitted to the RMSE scores to create a trend line to see the changes in network performance across iterations.

## 2.5 Statistical and video analyses

Statistical analyses were carried out in Rstudio 4.0.3 (R development Core Team, 2021). The data was tested for normality and did not meet assumptions for parametric tests. Visual inspection of the data revealed a Poisson distribution with counts of RMSE scores being independent from one another. Once networks had been tested against the training datasets to validate their performance, a generalised linear model (GLM) with a Poisson distribution was performed, using RMSE scores as a response variable to the interaction between network type and iterations. A critical significance value of 0.05 was chosen to find out how the magnitude of changes in performance varied between networks.

After finding the number of iterations where the performance of each network was best, a Kruskall-Wallis test was conducted on the RMSE scores and post-hoc pairwise Wilcoxon tests were used

to find out if one network outperformed others in the training. With videos from both camera setups, the network with the best overall training result was then used to analyse the validation videos. The 'deeplabcut.plot_trajectories()' function was used to generate key figures to count the number of bumblebees identified across these videos, by counting the maximum number of single body parts for each individual bumblebee in the figures. A Spearmann's correlation test was then conducted on these bumblebee counts for both camera setups to compare counts recorded after watching the videos in real time to the counts obtained from DLC in order to see for which setup were the networks better at identifying bumblebees.

## Results

### 3.1 Training results for different iterations between networks

For networks trained with the tube videos, the GLM with Poisson error structure revealed a significant difference in the pixel errors measured in RMSE scores between iterations, showing that iterations significantly decrease RMSE scores with increasing iterations and therefore improved the performance of these networks (f(20781,20776) = -18.145, p= 2e-16, **table A2**): [f(df1,df2) = z-value, p value]. The GLM also revealed a significant difference in RMSE scores between Dlcrnet_ms5 and ResNet50 networks (f(20781,20776) = -4.607, p= 4.09e-06, **table A2**) with the latter being significantly less performant. No significant difference was found between Dlcrnet_ms5 and ResNet50 (f(20781,20776) = 0.688, p= 0.491, **table A2**) indicating a similar performance between these networks. The GLM results of the interaction between network type and iterations revealed a significant difference in the magnitude of change in RMSE scores between Dlcrnet_ms5 and ResNet50 (f(20781,20776) = 16.366, p= 2e-16, **table A2**) with significantly higher changes for ResNet50. This suggest that iterations affect these networks differently. This was not the case for RMSE changes between ResNet50 and Dlcrnet_ms5 (f(20781,20776) = 0.765, p= 0.445, **table A2**).

For the nest entrance footage the RMSE scores were not found significantly different between Dlcrnet_ms5 and ResNet50 (f(3665,3660)) = 1.837, p= 0.0662, **table A3**) or between Dlcrnet_ms5 and ResNet101 (f(3665,3660) = 0.839, p= 0.4012, **table A3**). However, the magnitude of RMSE change between iterations within networks was significantly greater for both ResNet50 (f(3665,3660) = 2.071, p= 0.0384, **table A3**) and ResNet101 (f(3665,3660) = 2.312, p= 0.0208, **table A3**) compared to Dlcrnet_ms5, with ResNet101 having the greatest magnitude of change. Like the tube videos, iterations also significantly decreased RMSE scores with increasing iterations (f(3665,3660) = -2.117, p= 0.034283, **table A3**).

### 3.2 No strong fluctuation in performance between neural networks

For the trainings on tube and nest entrance videos there is no high deviation in the accuracy of body part predictions between networks, as RMSE scores do not group around a specific iteration number. With the default parameters, ResNet50 appears to achieve similar performance with lower and higher iterations (**table A4**). Overall, the lowest summed RMSE scores for this network was at 19,000 iterations with a score of 1,128.035 and the smallest score for a single body part was at 9,000 iterations with 0.033 (**table A4**). Across all predictions the highest error for a single body part was at 11,000 iterations with a score of 26.039 (**table A4**). No strong fluctuation in performance with increasing iterations can be observed for this network (**figure 4**).

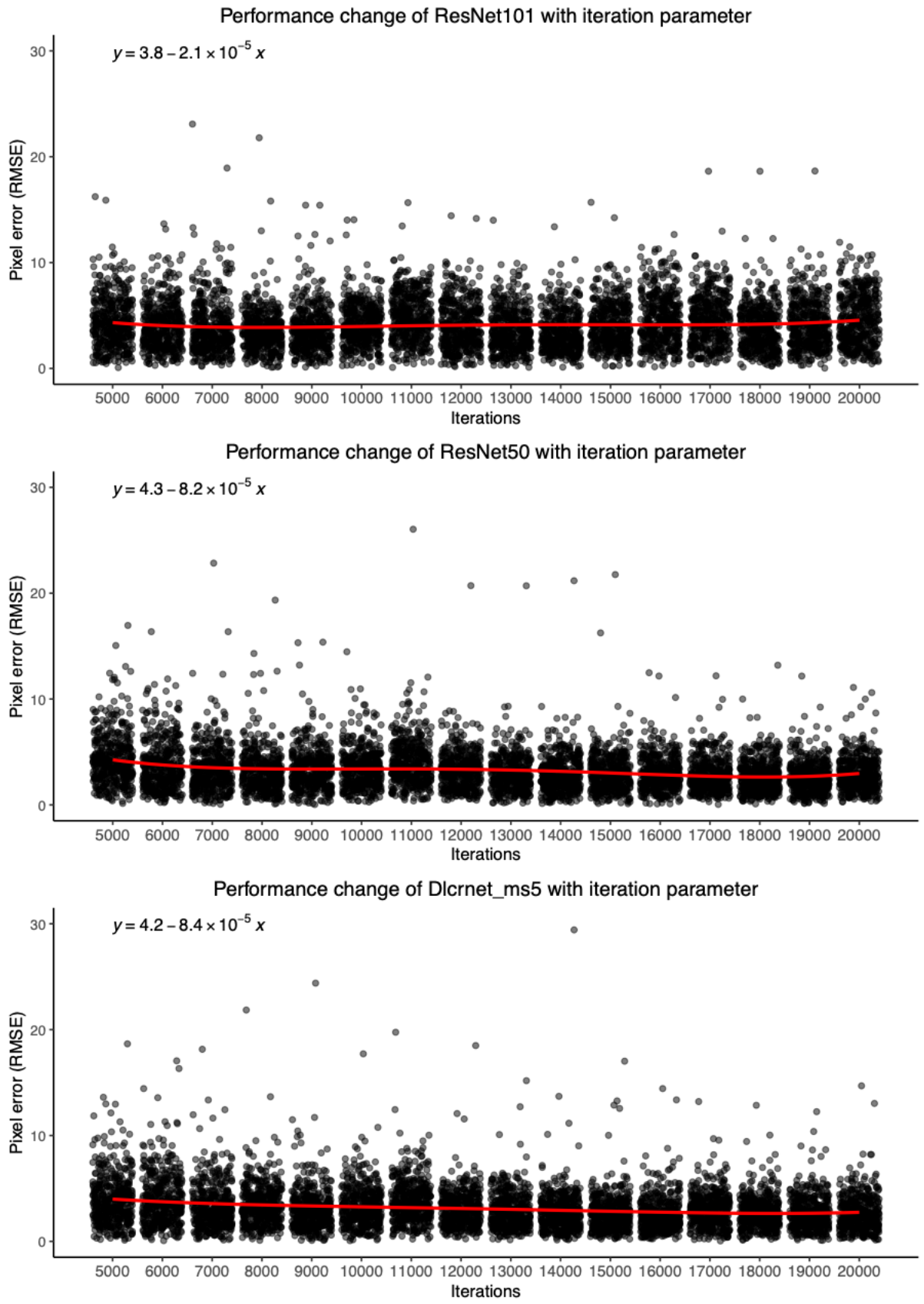
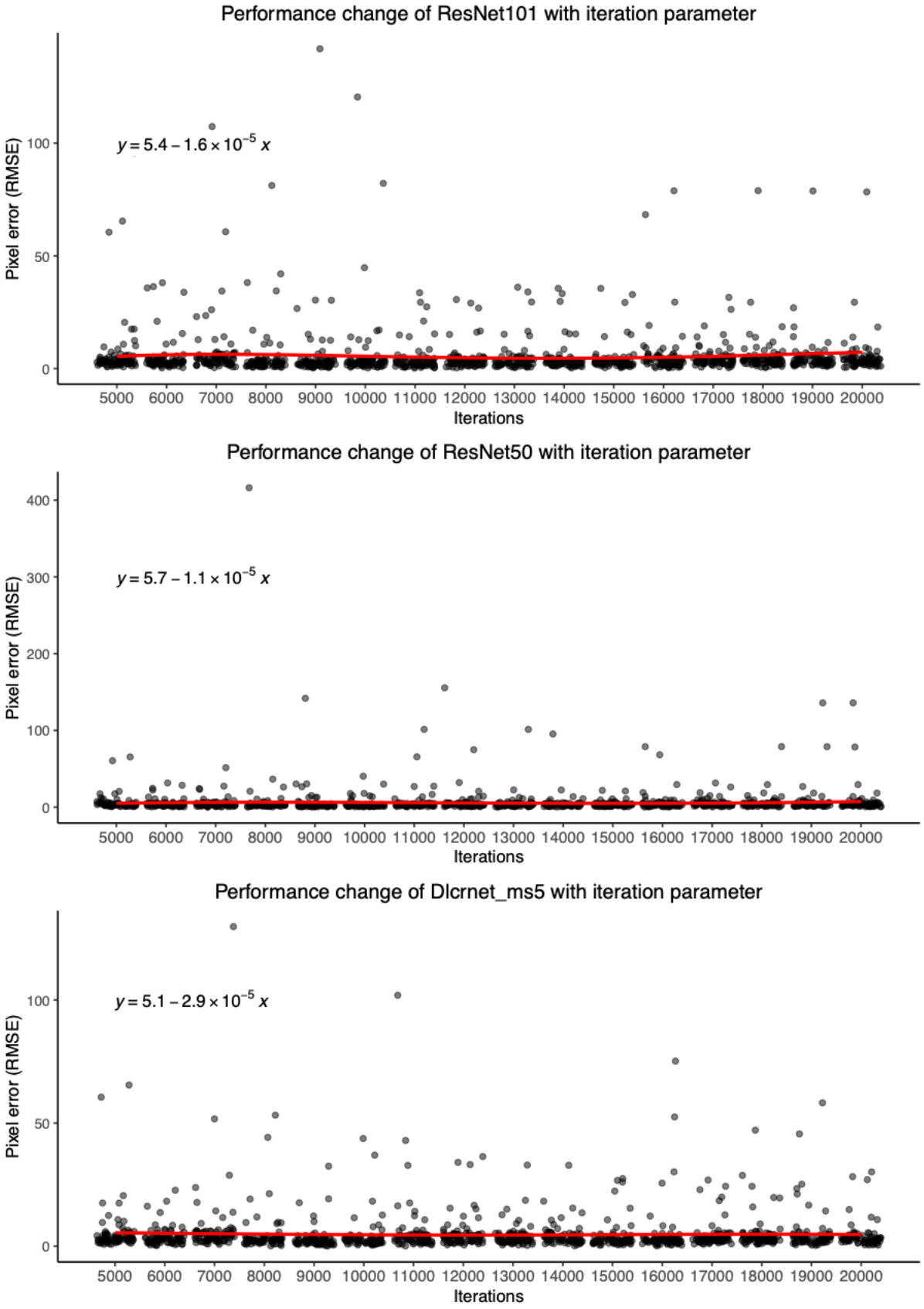



**Figure 4.** Residual mean squared error scores of three neural networks trained on tube videos across sixteen different iterations with 433 data points per iteration. Red line showing performance trend and that Dlcrnet_ms5 has the steepest negative slope and thereby the highest performance increase across iterations.

With the default parameters, ResNet_101 appears to achieve similar performance across iterations (**table A5**). The lowest summed RMSE scores of this network was at 14,000 iterations with 1,494.51. The smallest single RMSE score for a body part was at 19,000 iterations with 0.029 and the highest for a body part was at 7,000 iterations with 23.083 (**table A5**). By looking at the performance trend (**figure 4**) there is no strong RMSE decrease after 5,000 iterations. For Dlcrnet_ms5, the lowest summed RMSE scores was at 16,000 iterations with 1,122.016 and the smallest single RMSE score for a body part at 17,000 iterations with 0.04 (**table A6**). The highest error for a single prediction was at 14,000 iterations with 29.421. Also, looking at the equations of the fitted trend lines (**figure 4**), Dlcrnet_ms5 has the steepest negative slope and thereby the most performance increase with increasing iterations. For each network type the iteration with the best training performance, where the summed RMSE score is lowest, was selected to compare networks. Selected optimised networks for trainings on tube videos were: ResNet50 trained with 19,000 iterations, ResNet101 trained with 14,000 iterations and Dlcrnet_ms5 with 16,000 iterations.

For the nest entrance videos, the networks showed similar performance across iterations with the default parameters. Overall, the lowest summed RMSE scores for ResNet50 was at 15,000 with 284.854 and the same iteration also had the lowest summed RMSE for ResNet101 with 337.908 (**tables A7 & A8**). The highest errors for single body part predictions were at 8000 iterations with 416.202 for ResNet50 and at 9,000 with 141.894 for ResNet101 (**tables A7 & A8**). Dlcrnet_ms5 had the lowest summed RMSE at 9,000 iterations with 236.66 and a highest error at 7,000 with 129.785 (**table A9**). When looking at the performance trend of these networks no strong fluctuation can be observed with increasing iterations, however, Dlcrnet_ms5 again had the steepest negative slope indicating a greater performance increase across iterations (**figure 5**). Based on the lowest summed RMSE scores across training iterations, the selected optimised networks for nest entrance videos were: Dlcrnet_ms5 trained with 16,000 iterations and both ResNet50 and ResNet101 trained with 15,000 iterations.




256

**Figure 5.** Residual mean squared error scores of three neural networks trained on nest entrance videos across sixteen different iterations with 75 data points per iteration. Red line showing performance trend with Dlcrnet_ms5 having the steepest negative slope and thereby the highest performance increase across iterations.

### 3.3 Dlcrnet_ms5 outperformed ResNet50 & ResNet101

The Kruskall-Wallis results for the tube videos indicate that medians of the RMSE scores between optimised networks are significantly different ($X^2$=66.762, n=433, df=2, p = 3.183e-15). Post-hoc pairwise Wilcoxon tests revealed a significant difference between RMSE scores of Dlcrnet-ms5 and ResNet101 (W=25328, n=433, p= 9.56e-17, **figure 6A**) but not between Dlcrnet-ms5 and ResNet50 (W=47774, n=433, p= 0.761, **figure 6A**). RMSE scores were also found significantly difference between ResNet101 and ResNet50 (W=67317, n=433, p= 5.95e-15, **figure 6A**). Therefore, these networks do compare in the accuracy of their body part predictions.

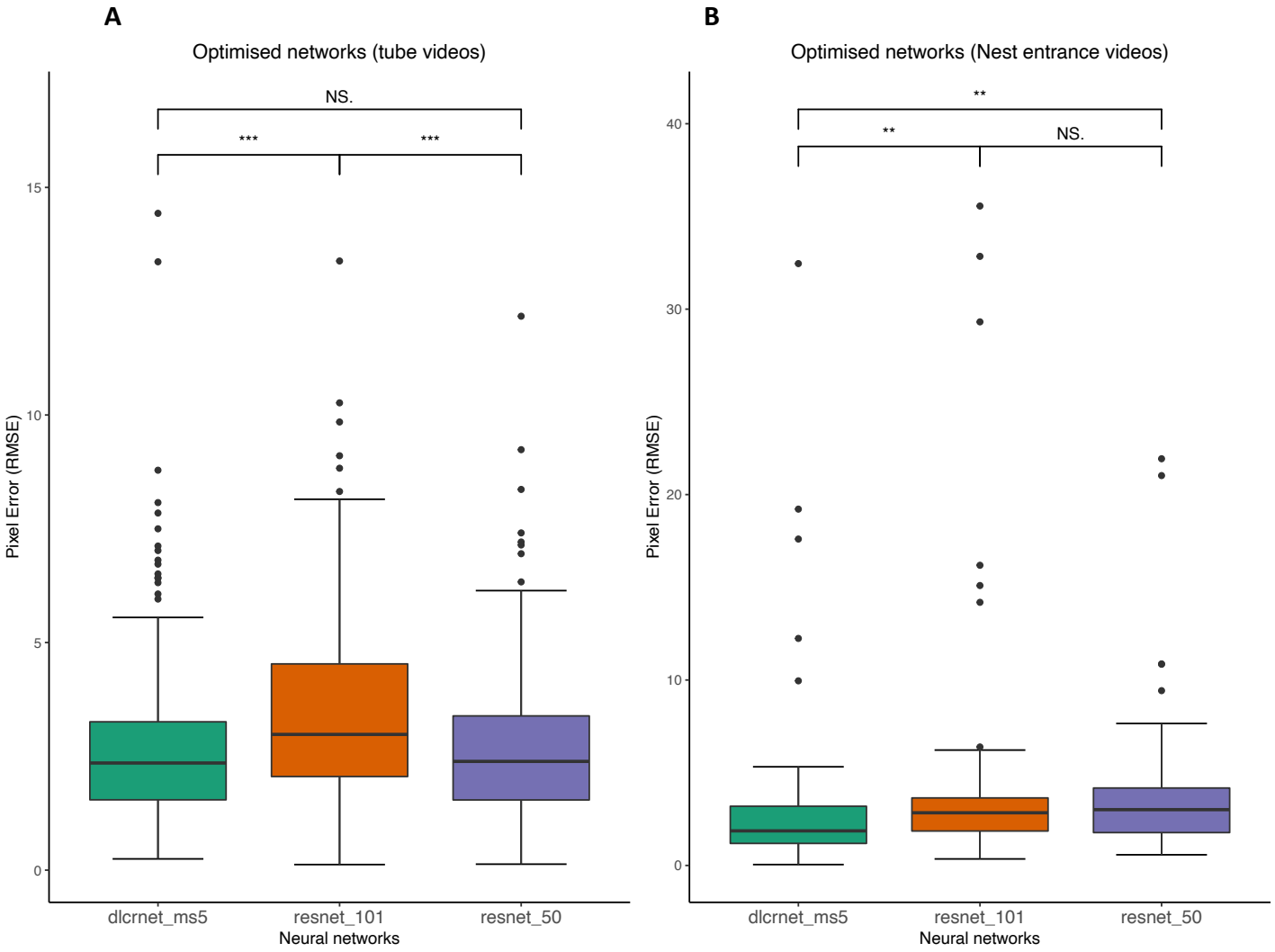



**Figure 6. A**) Boxplots comparing medians of residual mean squared errors of 3 neural networks optimised with their best iteration/training performance on tube videos, with 433 annotation points per network. **B**) Boxplots comparing medians of residual mean squared errors of 3 neural networks optimised with their best iteration/training performance on nest entrance videos, with 75 annotation points per network.

ResNet101 was also worst at predicting body parts for tube videos as it had a higher RMSE mean value (**table A5**). By comparing RMSE mean scores of these optimised networks, Dlcrnet_ms5 has the smallest value (**tables A4 & A6**) and was therefore chosen to analyse the validation tube videos as it made more accurate predictions during training.

Kruskall-Wallis results for the networks trained on nest entrance videos also revealed a significant difference in the medians of RMSE scores between networks ($X^2$=12.325, n=433, df=2, p = 0.002107). Post-hoc pairwise Wilcoxon tests revealed a significant difference between RMSE scores of Dlcrnet-ms5 and ResNet101 (W=2108, n=75, p= 0.004, **figure 6B**) and between RMSE scores of Dlcrnet-ms5 and ResNet50 (W=1976, n=75, p= 0.002, **figure 6B**). No significant difference was found in scores between ResNet50 and ResNet101 (W=2717, n=75, p= 0.531, **figure 6B**). Dlcrnet_ms5 was selected to analyse the validation nest entrance videos as it outperformed other networks during training by having the smallest RMSE mean (**table A9**) and was thereby more accurate at making body part predictions.

### 3.4 Analyses of videos

Tube videos were analysed with Dlcrnet_ms5 trained on 16,000 iterations and the same network trained on 9,000 iterations was used to analyse nest entrance videos. The Spearman's correlation tests revealed positive corelations between the human and DLC counts for the tube videos (R= 0.74, n= 15, p=0.0017, **figure 7A**) and for the nest entrance videos (R= 0.56, n= 15, p=0.032, **figure 7B**). Looking at the correlation coefficients, a stronger correlation was found between counts with the tube videos (**figure 7A**) suggesting that DLC is better able to identify bumblebees in recordings with less occlusions. A summary of counts recorded by human observation and DLC for both validation datasets can be found in **table A12**.

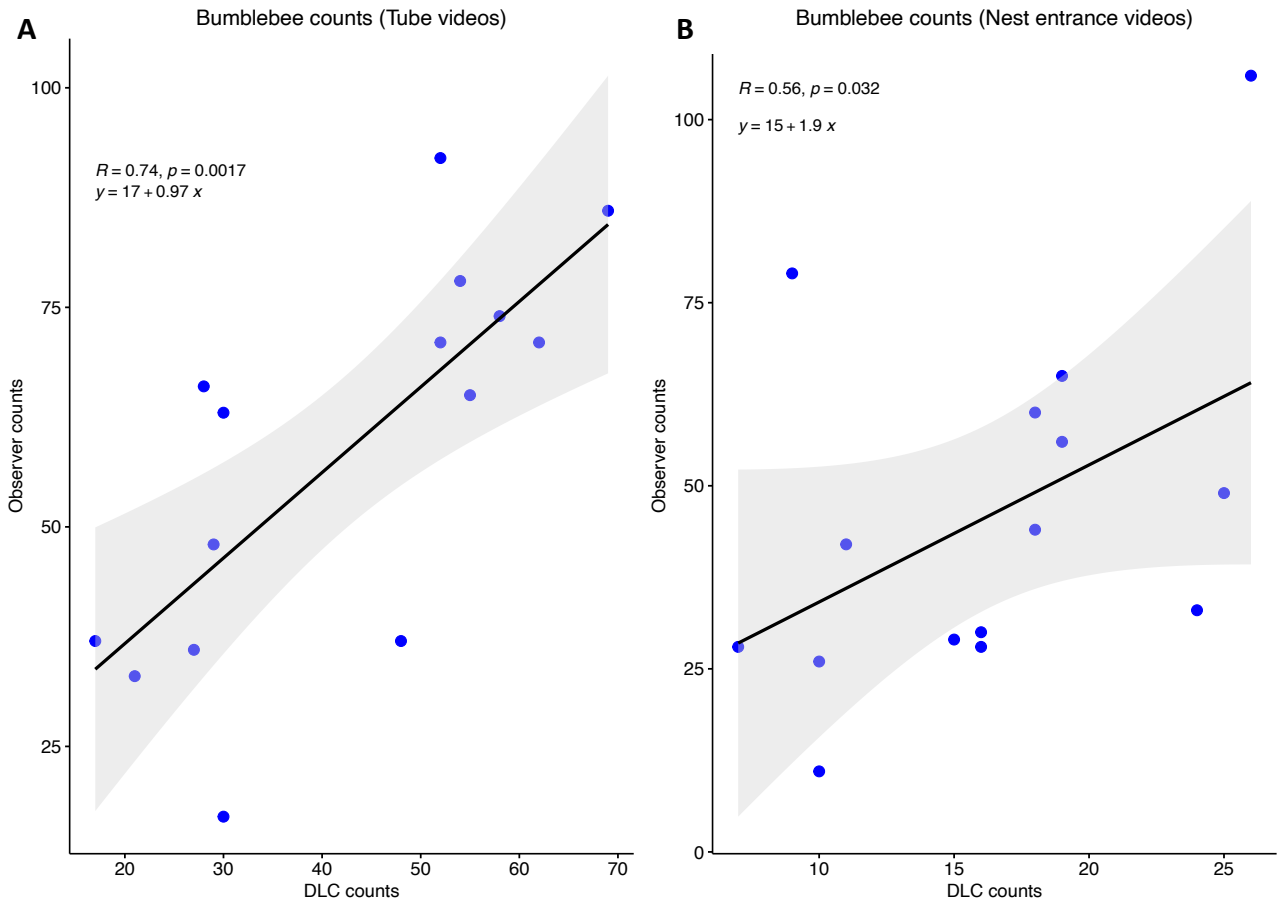



**Figure 7.** Scatterplots of bumblebee counts from human observation and DLC counts showing positive correlations for both tube videos (n=15) and nest entrance videos (n=15). A stronger correlation can be seen between counts made on the tube videos with less occlusions than counts made on the nest entrance videos with more occluded scenes.

## Discussion

DeepLabCut, a state of the art pose estimation tool was used to track bumblebees in an open field experiment with two camera setups and count bumblebees within videos. As seen in the results, three neural networks (ResNet50, ResNet101 and Dlcrnet_ms5) were able to track bumblebees and Dlcrnet_ms5 had the smallest amount of pixel errors with both camera setups and was therefore selected for video analyses. Moreover, the iteration parameter was found to significantly decrease pixel errors with increasing iterations and so increases the network performance at predicting body parts. However, no saturation curve with a plateau was observed as there was no strong decrease in pixel errors with increasing iterations. The decrease in pixel errors with increasing iterations was greater for Dlcrnet_ms5 than other networks and was not consistent between networks. Therefore, the training of these networks was optimised at different iterations. It was also found that the accuracy of Dlcrnet_m5 at tracking bumblebees in videos with more occlusions was worse than in videos with fewer occlusion scenes.

The hypothesis that neural networks can be trained to track bumblebees from video recordings was confirmed as all three networks were able to make predictions for every annotated body part. Moreover, the utilisation of three neural networks showed that different networks perform relatively well at predicting body parts but Dlcrnet_ms5 made more accurate predictions compared to ResNet50 and ResNet10. ResNet101 is considered an improvement of ResNet50, but here I found it to deliver the lowest accuracy on tube videos compared to ResNet50 but a better accuracy than ResNet50 on the nest entrance videos with more occlusions. Dlcrnet_ms5 is a more recent and specialised network that is an implement of DeeperCut made specifically to detect animal movement (Lauer et al., 2021). This network was also designed for tracking animals from multi-animal videos which can explain why its predictions were better than ResNet101 & ResNet50 that have a more general design for image recognition (Feng, 2017 ; Lauer et al., 2021).

The iteration parameter did increase the performance of networks suggesting that more iterations improve training performance. However, higher iterations only decreased pixel errors smally by -2.689e-05 RMSE for predictions of the tube videos and by -6.008e-06 RMSE for nest entrance videos. This explains why no saturation curve with a plateau could be observe with increasing iterations and RMSE scores. Additional iterations could significantly improve the accuracy of predictions as supported by other studies (Labuguen et al., 2019 ; Clemensson et al., 2020). Therefore the second hypothesis that the tracking performance increases with more training iterations was supported. As iterations are a core parameter that adjust the training weights to be closer to expected pixel values, more iterations are expected to improve accuracy at the cost of computational power and time. Iterations impact the learning rates of neural networks (Abbas, Bangyal and Ahmad, 2013) affecting the accuracy of body part predictions. The three networks used are recommended to be trained around 50,000 to 100,000 iterations with the default batch size of 8 (Lauer et al., 2021). With more computing resources, future experiments could investigate the training performance of these networks on more iterations.

The last hypothesis that neural networks are better able to track bumblebees from videos with less occlusions was confirmed. DLC counts were closer to the human observer counts in the videos of bumblebees walking through clear tubes compared to the videos with more occlusions of bumblebees flying around the nest entrance. However, the tube videos had more annotations than the nest entrance videos (433 vs 75) which could be why DLC was better able to track bumblebees in this setup. A better approach would have been to standardize the number of annotations for both camera setups to make them more comparable. Also, two of the validation nest entrance videos were out of focus which could explain why predictions were worse. The maximum number of bumblebees expected in an image was predefined to 10 individuals in the configuration file. This

was not enough as the validation videos for both datasets had scenes with over 10 bumblebees, meaning that DLC was unable to track additional bumblebees. This could explain why overall DLC counts for both camera setups were less than human counts.

As all videos in this study were recorded with same cameras settings, in the future it would be interesting to compare network performances with different settings as this would have affected labelling and therefore the accuracy of predictions. Also, because the distance between the cameras and the tubes or nest entrance were not standardized between recordings, RMSE scores were not comparable between each video. Furthermore, the networks compared in this study were pretrained on ImageNet. This is known as transfer learning, the ability to use a network for a task on a small supervised dataset that was previously trained on another task with a larger supervised dataset (Nath et al., 2019). With transfer learning, deeplearning methods can be applied by more users in their research and there is an increasing use of pretrained neural networks for animal tracking (Ratnayake, Dyer and Dorin, 2021). These networks are also built on a similar type of architecture that allows to track objects in frames irrespective of the background to reach high accuracy levels depending on the quality of training data (Feng, 2017 ; Ratnayake, Dyer and Dorin, 2021).

On the quality of training in this study, the training datasets only took a few days to annotate and had fewer videos than the validation datasets with fewer bumblebees. More annotated bumblebees would have improved the network performances. Also, the validation videos for the tube videos had occlusions that were not present in the training videos. The training quality could have been improved with more variability added to the training by accounting for more social behaviours. To improve training quality DLC recommends using image augmentation with the imgaug library (Mathis et al., 2020) which was not used in this study. Augmentations are means of changing images to increase the amount of variability in training to improve network performance (Perez and Wang 2017). Results of this study would have been different with more training videos, but with limitations due to time constraints and maintenances on the HPC it was decided to only alter the iteration parameter to improve training.

Collecting behavioural data is challenging and the study of insect pollination is one field where fine-scale behavioural data is lacking and difficult to collect as it is hard to monitor and track insect pollinators (Ratnayake, Dyer and Dorin, 2021). With the use of neural networks with computer vision tools, establishing field stations to record pollinator behaviour could offer a low cost, reliable and continual approach for monitoring populations. Over the past few years, pollinators including bumblebees (*Bombus spp.*) have shown sharp declines in population size across Europe

(Biesmeijer et al., 2006 and Potts et al., 2011). Bumblebees provide crucial ecosystem services and maintain plant diversity (Ollerton, Winfree and Tarrant, 2011). Among generalist pollinators, bumblebees rank amongst the top crop pollinators (Leonhardt and Blüthgen, 2022) and it is possible to monitor populations by looking at their foraging activity (Judd et al., 2020). About 70% of leading food crops around the world are dependent on pollinators (Klein et al., 2007). Therefore, ensuring the conservation of pollinators like bumblebees is an important factor towards stabilizing global food security and artificial neural networks could play a key role in this. Neural networks could also potentially be used for assessing the foraging quality of pollinators by looking at their pollen baskets to quantify the amount of pollen collected. This could be a better approach than current invasive methods with pollen traps that are labour intensive and remove some of the collected pollen that workers cannot access anymore, which can be deadly for small colonies that need more resources (Webster et al., 1985 ; Judd et al., 2020).

DLC is a toolbox for tracking animals that is moderately easy to use for forming datasets, training different types of neural networks, evaluating and using them to analyse videos. Other than annotating frames and adjusting labels there is no difficult manual labour required but one must have a laptop with a GPU in order to speed up the network training process with more iterations. In their original paper (Mathis et al., 2018) trained a network with 500,000 iterations in 24 hours using a GTX 1080 but as GPUs were unavailable in this study the networks could not been trained on larger iterations which would exceed the 72 hours HPC time limit. DLC is also actively developed with new features being introduced to improve its accuracy and suitability for wider usage. Recently DeepLabCut-Live! was released for real-time tracking (Kane et al. 2020) and authors have made an R script DLCAnalyzer for analysing tracking results (Shenk et al., 2021). DLC is said to be more robust and straightforward at analysing social behaviour of animals than other tracking tools (Lauer et al., 2021) and outperforms commercial tracking software (Sturman et al., 2020). With technological advances there are now more types of neural networks specialized in animal tracking (Ziegler, Sturman, and Bohacek 2021). Other well-known animal tracking tools include DeepPoseKite (Graving et al., 2019), DeepBehaviour, SLEAP (Pereira et al., 2020), DeepBehaviour and The Animal Part Tracker (Arac et al., 2019) which all use different algorithms for tracking body parts. As the accuracy of pose estimation is greatly improving with the evolution of these tools (Ziegler, Sturman, and Bohacek 2021) it would be interesting to compare the performance at tracking bumblebees of other tools than DLC to know which is closest to being fully automated for studying pollinator behaviour.

**Conclusion:** This project showed that deep neural networks can be used for studying fast moving insects like bumblebees. Different neural networks can be used to identify bumblebees but depending on how their architecture is built, it was found that some networks are better able to track bumblebees than others. The performance of networks at identifying bumblebees can also be improved by training networks on larger iterations. When it comes to occlusions in videos, neural networks are better able to track bumblebees in videos with fewer occlusions than videos with more occluded scenes. Tracking bumblebees from videos with neural networks is less time-consuming and error prone than if done by human observation. Tools like DeepLabCut can collect valuable pollinator behavioural data which could help in the protection and study of pollinator populations facing decline. Future work using DeepLabCut could be improved by considering the limitations affecting the quality of training in this study. I hope this project sparks the interest of other students, even those with little experience in machine learning to start using tools like DeepLabCut in their research.

## References

AlDuwaile, D.A. and Islam, M.S., 2021. Using convolutional neural network and a single heartbeat for ECG biometric recognition. *Entropy*, 23(6), p.733.

Arac, Ahmet, Pingping Zhao, Bruce H Dobkin, S Thomas Carmichael, and Peyman Golshani. 2019. "DeepBehavior: A deep learning toolbox for automated analysis of animal and human behavior imaging data". *Frontiers in systems neuroscience,* 13(20).

Arent, I., Schmidt, F., Botsch, M. and Dürr, V., 2021. Marker-Less Motion Capture of Insect Locomotion With Deep Neural Networks Pre-trained on Synthetic Videos. *Frontiers in Behavioral Neuroscience*, 15.

baeldung., 2020. *The Difference Between Epoch and Iteration in Neural Networks | Baeldung on Computer Science*. [online] www.baeldung.com. Available at: https://www.baeldung.com/cs/neural-networks-epoch-vs-iteration.

Biesmeijer, J.C., 2006. Parallel Declines in Pollinators and Insect-Pollinated Plants in Britain and the Netherlands. *Science*, 313 (5785), pp. 351–354.

Carvell, C., Westrich, P., Meek, W.R., Pywell, R.F. and Nowakowski, M., 2006. Assessing the value of annual and perennial forage mixtures for bumblebees by direct observation and pollen analysis. *Apidologie*. 37 (3), pp. 336-340.

Clemensson, E.K., Abbaszadeh, M., Fanni, S., Espa, E. and Cenci, M.A., 2020. Tracking Rats in Operant Conditioning Chambers Using a Versatile Homemade Video Camera and DeepLabCut. JoVE (Journal of Visualized Experiments), (160), p.e61409.

Dell, A.I., Bender, J.A., Branson, K., Couzin, I.D., de Polavieja, G.G., Noldus, L.P., Pérez-Escudero, A., Perona, P., Straw, A.D., Wikelski, M. and Brose, U., 2014. Automated image-based tracking and its application in ecology. Trends in ecology & evolution, 29(7), pp.417-428.

Feng, V., 2017. An overview of resnet and its variants. *Towards data science*, 2.

Graving, J.M., Daniel, C., Hemal, N., Liang, Li., Benjamin, K., Blair, R.C., and Iain, D.C., 2019. ”DeepPoseKit, a software toolkit for fast and robust animal pose estimation using deep learning”. *Elife* 8.

Hardin, A. and Schlupp, I., 2022. Using machine learning and DeepLabCut in animal behavior. *acta ethologica*, pp.1-9.

He, K., Xiangyu Z., Shaoqing R., and Jian S., 2016. Deep Residual Learning for Image Recognition. In Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition (CVPR).

He, K., Zhang, X., Ren, S. and Sun, J., 2016. Deep residual learning for image recognition. In Proceedings of the IEEE conference on computer vision and pattern recognition, pp. 770-778.

Janisch, J., Mitoyen, C., Perinot, E., Spezie, G., Fusani, L. and Quigley, C. (2021). Video Recording and Analysis of Avian Movements and Behavior: Insights from Courtship Case Studies. *Integrative and Comparative Biology*, 61(4), pp.1378–1393. doi:10.1093/icb/icab095.

Judd, H. J., Huntzinger, C., Ramirez, R., Strange, J. P., 2020. A 3D Printed Pollen Trap for Bumble Bee (Bombus) Hive Entrances. J. Vis. Exp. (161), e61500, doi:10.3791/61500.

Kane, G.A, Gonçalo, L., Jonny, L.S., Alexander, M., and Mackenzie, W.M., 2020. ”Real-time, low-latency closed-loop feedback using markerless posture tracking”. *Elife,* 9:e61909.

Klein, A. M., Vaissiere, B. E., Cane, J. H., Steffan-Dewenter, I., Cunningham, S. A., Kremen, C., & Tscharntke, T., 2007. Importance of pollinators in changing landscapes for world crops. Proceedings. *Biological Sciences / the Royal Society*, 274 (1608), pp. 303-313.

Koushik, J., 2016. Understanding convolutional neural networks. *arXiv* preprint arXiv:1605.09081.

Labuguen, R., Bardeloza, D.K., Negrete, S.B., Matsumoto, J., Inoue, K. and Shibata, T., 2019, May. Primate markerless pose estimation and movement analysis using DeepLabCut. In 2019 Joint 8th International Conference on Informatics, Electronics & Vision (ICIEV) and 2019 3rd International Conference on Imaging, Vision & Pattern Recognition (icIVPR) (pp. 297-300). IEEE.

Lauer, J., Zhou, M., Ye, S., Menegas, W., Nath, T., Rahman, M.M., Di Santo, V., Soberanes, D., Feng, G., Murthy, V.N., Lauder, G., Dulac, C., Mathis, M.W. and Mathis, A., 2021. Multi-animal pose estimation and tracking with DeepLabCut.

Lauer, J., Zhou, M., Ye, S., Menegas, W., Schneider, S., Nath, T., Rahman, M., Di Santo, V., Soberanes, D., Feng, G., Murthy, V., Lauder, G., Dulac, C., Mathis, M. and Mathis, A., 2022. Multi-animal pose estimation, identification and tracking with DeepLabCut. *Nature Methods*, 19(4), pp.496-504.

Leonhardt, S. and Blüthgen, N., 2022. The same, but different: pollen foraging in honeybee and bumblebee colonies.

Lou, G. and Shi, H., 2020. Face image recognition based on convolutional neural network. China Communications, 17(2), pp.117-124.

Mathis, A., Mamidanna, P., Cury, K.M., Abe, T., Murthy, V.N., Mathis, M.W. and Bethge, M., 2018. DeepLabCut: markerless pose estimation of user-defined body parts with deep learning. *Nature Neuroscience*, [online] 21(9), pp.1281–1289.

Mathis, Alexander, Steffen Schneider, Jessy Lauer, and Mackenzie Weygandt Mathis., 2020. A Primer on Motion Capture with Deep Learning: Principles, Pitfalls, and Perspectives. *Neuron*, 108 (1): pp.44–65.

Nath, A., Mathis, A., Chen, C.A., Patel, A., Bethge, M. and Mathis, W.M., 2019. Using DeepLabCut for 3D markerless pose estimation across species and behaviours. *Nature Protocols*, (14), pp. 2152-2176.

Nielsen, M. A.,2018. Neural Networks and Deep Learning. *Determination Press*.

Ollerton, J., Winfree, R., and Tarrant, S., 2011. How many flowering plants are pollinated by animals? *Oikos*. 120 (3), pp. 321-326.

Pereira, T.D., Nathaniel, T., Junyu, L., Shruthi, R., Eleni, S,P,. Z, Y.W., David, M,T, Grace, McK., Sarah, D.K., and Annegret, L.F., 2020. SLEAP: Multi-animal pose tracking. *BioRxiv*.

Perez, L., and Jason, W., 2017. The effectiveness of data augmentation in image classification using deep learning. *arXiv,* preprint arXiv:1712.04621.

Potts, S.G., Biesmeijer, J.C., Kremen, C., Neumann, P., Schweiger, O. and Kunin, W.E. (2010). Global pollinator declines: trends, impacts and drivers. *Trends in Ecology & Evolution*, 25 (6), pp. 345–353.

Ratnayake, M., Dyer, A. and Dorin, A., 2021. Tracking individual honeybees among wildflower clusters with computer vision-facilitated pollinator monitoring. *PLOS ONE*, 16(2), p.e0239504.

RStudio Team, 2015. RStudio: Integrated Development Environment for R, Boston, MA. Available at: http://www.rstudio.com/.

Rodriguez, A., Zhang, H., Klaminder, J., Brodin, T. and Andersson, M., 2017. ToxId: an efficient algorithm to solve occlusions when tracking multiple animals. *Scientific Reports*, 7(1).

Sclocco, A., Ong, S., Pyay Aung, S. and Teseo, S., 2021. Integrating real-time data analysis into automatic tracking of social insects. *Royal Society Open Science*, 8(3).

Shenk, J., Byttner, W., Nambusubramaniyan, S. and Zoeller, A., 2021. Traja: A Python toolbox for animal trajectory analysis. *Journal of Open Source Software*, *6*(63), pp.3202.

Sturman, Oliver, Lukas von Ziegler, Christa Schläppi, Furkan Akyol, Mattia Privitera, Daria Slominski, Christina Grimm, 2020. Deep learning-based behavioral analysis reaches human accuracy and is capable of outperforming commercial solutions. *Neuropsychopharmacology,* 45 (11), pp.1942–1952.

Traner, M., Chandak, R. and Raman, B., 2021. Recent approaches to study the neural bases of complex insect behavior. *Current Opinion in Insect Science*, 48, pp.18-25.

Webster, T.C., Thorp, R.W., Briggs, D., Skinner, J. and Parisian, T., 1985. Effects of pollen traps on honey bee (Hymenoptera: Apidae) foraging and brood rearing during almond and prune pollination. *Environmental Entomology*, 14(6), pp.683-686.

Ziegler, L., Oliver, S., and Johannes, B., 2021. Big behavior: challenges and opportunities in a new era of deep behavior profiling. *Neuropsychopharmacology*, 46 (1),pp. 33–44.

Zur, R.M., Jiang, Y. and Metz, C.E., 2004, June. Comparison of two methods of adding jitter to artificial neural network training. *International Congress Series,* (1268), pp. 886-889).

## Appendices

**Table A1.** Resources used on the Imperial HPC cluster for the training of three neural networks.

| *Network type* | *Iterations* | *Number of CPUs* | *Memory GB* | *Wall time (hours)* |
|---|---|---|---|---|
| *Dlcrnet-ms5* | 5000 to 10000 | 16 | 900 | 14 |
| *Dlcrnet-ms5* | 11000 to 15000 | 36 | 2000 | 28 |
| *Dlcrnet-ms5* | 16000 to 20000 | 70 | 3500 | 72 |
| *ResNet_50* | 5000 to 10000 | 16 | 900 | 14 |
| *ResNet_50* | 11000 to 15000 | 36 | 2000 | 28 |
| *ResNet_50* | 16000 to 20000 | 70 | 3500 | 72 |
| *ResNet_101* | 5000 to 10000 | 16 | 900 | 14 |
| *ResNet_101* | 11000 to 15000 | 36 | 2000 | 28 |
| *ResNet_101* | 16000 to 20000 | 70 | 3500 | 72 |

**Table A2.** Outputs of the generalised linear model with Poisson distribution and log-link function of the RMSE scores (n= 20,784) of three neural networks trained on the tube videos with different iterations, null deviance: 24449 on 20781 df and residual deviance: 22662 on 20776 df ; calculated pseudo $R^2$=0.07307754.

| *Coefficients* | *Estimate* | *Standard error* | *Z value* | *P value* |
|---|---|---|---|---|
| *Dlcrnet_ms5 (intercept)* | 1.467 | 1.895e-02 | 77.411 | **2e-16*** ** |
| *ResNet50* | -1.181e-01 | 2.563e-02 | -4.607 | **4.09e-06*** ** |
| *ResNet101* | 1.829e-02 | 2.657e-02 | 0.688 | 0.491 |
| *Iterations* | -2.689e-05 | 1.482e-06 | -18.145 | **2e-16*** ** |
| *ResNet50 : Iterations* | 3.211e-05 | 1.962e-06 | 16.366 | **2e-16*** ** |
| *ResNet101 : Iterations* | 1.588e-06 | 2.076e-06 | 0.765 | 0.445 |

**Table A3.** Outputs of the generalised linear model with Poisson distribution and log-link function of the RMSE scores (n= 3,600) of three neural networks trained on the nest entrance videos with different iterations, null deviance: 30927 on 3665 df and residual deviance: 30790 on 3660 df ; calculated pseudo $R^2$=0.004431659.

| *Coefficients* | *Estimate* | *Standard error* | *Z value* | *P value* |
|---|---|---|---|---|
| *Dlcrnet_ms5 (intercept)* | 1.640 | 3.737e-02 | 43.903 | **2e-16*** ** |
| *ResNet50* | 9.325e-02 | 5.076e-02 | 1.837 | 0.0662 |
| *ResNet101* | 4.296e-02 | 5.118e-02 | 0.839 | 0.4012 |
| *Iterations* | -6.008e-06 | 2.839e-06 | -2.117 | **0.034283*** |
| *ResNet50 : Iterations* | 7.938e-06 | 3.833-06 | 2.071 | **0.0384*** |
| *ResNet101 : Iterations* | 8.928e-06 | 3.862e-06 | 2.312 | **0.0208*** |

**Table A4**. Pixel error using RMSE scores from the ResNet_50 training on the camera facing the tubes and with 433 labelled body parts across 360 frames for each iteration.

| ***Sum RMSE*** | ***Mean RMSE*** | ***Highest RSME*** | ***Lowest RMSE*** | ***Iterations*** |
|---|---|---|---|---|
| *1877.835* | 4.34 | 16.953 | 0.139 | 5000 |
| *1625.585* | 3.75 | 16.363 | 0.195 | 6000 |
| *1494.094* | 3.45 | 22.838 | 0.059 | 7000 |
| *1375.199* | 3.17 | 19.347 | 0.093 | 8000 |
| *1446.091* | 3.34 | 15.373 | **0.033** | 9000 |
| *1514.737* | 3.5 | 14.464 | 0.272 | 10000 |
| *1760.13* | 4.064 | **26.039** | 0.378 | 11000 |
| *1420.802* | 3.281 | 20.712 | 0.447 | 12000 |
| *1272.518* | 2.938 | 20.698 | 0.176 | 13000 |
| *1255.31* | 2.899 | 21.173 | 0.133 | 14000 |
| *1287.512* | 2.973 | 21.754 | 0.109 | 15000 |
| *1294.423* | 2.989 | 12.485 | 0.096 | 16000 |
| *1228.359* | 2.836 | 12.194 | 0.044 | 17000 |
| *1182.6* | 2.731 | 13.197 | 0.095 | 18000 |
| ***1128.035*** | **2.605** | 12.171 | 0.131 | **19000** |
| *1278.275* | 2.95 | 11.097 | 0.292 | 20000 |

**Table A5**. Pixel error using RMSE scores from the ResNet_101 training on the camera facing the tubes and with 433 labelled body parts across 360 frames for each iteration.

| ***Sum RMSE*** | ***Mean RMSE*** | ***Highest RSME*** | ***Lowest RMSE*** | ***Iterations*** |
|---|---|---|---|---|
| *1893.078* | 4.372 | 16.228 | 0.078 | 5000 |
| *1762.161* | 4.069 | 13.664 | 0.284 | 6000 |
| *1705.4* | 3.938 | **23.083** | 0.201 | 7000 |
| *1536.692* | 3.549 | 21.785 | 0.111 | 8000 |
| *1608.121* | 3.714 | 15.422 | 0.129 | 9000 |
| *1794.279* | 4.144 | 14.042 | 0.187 | 10000 |
| *2120.717* | 4.909 | 15.659 | 0.097 | 11000 |
| *1762.211* | 4.079 | 14.442 | 0.167 | 12000 |
| *1586.311* | 3.664 | 13.999 | 0.109 | 13000 |
| ***1494.51*** | **3.452** | 13.383 | 0.122 | **14000** |
| *1749.221* | 4.04 | 15.691 | 0.221 | 15000 |
| *2014.092* | 4.651 | 12.652 | 0.157 | 16000 |
| *2002.335* | 4.624 | 18.631 | 0.391 | 17000 |
| *1636.807* | 3.780 | 18.627 | 0.131 | 18000 |
| *1805.045* | 4.168 | 18.649 | **0.029** | 19000 |
| *2011.352* | 4.645 | 11.914 | 0.439 | 20000 |

**Table A6**. Pixel error using RMSE scores from the Dlcrnet_ms5 network trained on the tube dataset of the camera facing the tubes and with 433 labelled body parts across 360 frames for each iteration.

| ***Sum RMSE*** | ***Mean RMSE*** | ***Highest RSME*** | ***Lowest RMSE*** | ***Iterations*** |
|---|---|---|---|---|
| *1714.568* | 3.959 | 18.659 | 0.114 | 5000 |
| *1723.197* | 3.979 | 17.053 | 0.067 | 6000 |
| *1440.475* | 3.326 | 18.145 | 0.13 | 7000 |
| *1490.959* | 3.443 | 21.853 | 0.096 | 8000 |
| *1331.323* | 3.075 | 24.393 | 0.092 | 9000 |
| *1420.394* | 3.280 | 17.715 | 0.065 | 10000 |
| *1644.103* | 3.797 | 19.754 | 0.218 | 11000 |
| *1331.692* | 3.075 | 18.493 | 0.233 | 12000 |
| *1251.419* | 2.890 | 15.180 | 0.145 | 13000 |
| *1222.032* | 2.822 | **29.421** | 0.136 | 14000 |
| *1157.768* | 2.673 | 17.019 | 0.223 | 15000 |
| ***1122.019*** | **2.591** | 14.430 | 0.246 | **16000** |
| *1265.149* | 2.921 | 13.204 | **0.04** | 17000 |
| *1197.525* | 2.765 | 12.854 | 0.118 | 18000 |
| *1155.355* | 2.668 | 12.252 | 0.093 | 19000 |
| *1152.93* | 2.662 | 14.691 | 0.129 | 20000 |

**Table A7**. Pixel error using RMSE scores from the ResNet50 training on the camera facing the nest entrance and with 75 labelled body parts across 360 frames for each iteration.

| ***Sum RMSE*** | ***Mean RMSE*** | ***Highest RSME*** | ***Lowest RMSE*** | ***Iterations*** |
|---|---|---|---|---|
| *445.212* | 5.707 | 65.44 | 0.696 | 5000 |
| *364.376* | 4.732 | 31.613 | 0.42 | 6000 |
| *391.653* | 5.222 | 51.482 | **0.128** | 7000 |
| *727.863* | 9.577 | **416.202** | 0.344 | 8000 |
| *445.597* | 5.787 | 141.894 | 0.301 | 9000 |
| *320.885* | 4.114 | 40.375 | 0.471 | 10000 |
| *475.23* | 6.172 | 101.355 | 0.498 | 11000 |
| *556.86* | 7.232 | 155.556 | 0.461 | 12000 |
| *368.43* | 4.912 | 101.323 | 0.241 | 13000 |
| *358.528* | 4.78 | 95.4 | 0.391 | 14000 |
| ***284.854*** | **3.798** | 21.933 | 0.576 | **15000** |
| *459.406* | 6.045 | 78.89 | 0.332 | 16000 |
| *459.407* | 4.987 | 31.589 | 0.4 | 17000 |
| *423.298* | 5.57 | 78.95 | 0.488 | 18000 |
| *533.679* | 7.022 | 135.879 | 1.016 | 19000 |
| *540.838* | 7.116 | 135.965 | 0.544 | 20000 |

**Table A8.** Pixel error using RMSE scores from the ResNet101 training on the camera facing the nest entrance and with 75 labelled body parts across 360 frames for each iteration.

| *Sum RMSE* | *Mean RMSE* | *Highest RSME* | *Lowest RMSE* | *Iterations* |
|---|---|---|---|---|
| *445.213* | 5.71 | 65.44 | 0.696 | 5000 |
| *396.362* | 5.148 | 38.081 | 0.307 | 6000 |
| *584.059* | 7.488 | 107.398 | 0.417 | 7000 |
| *415.779* | 5.4 | 81.244 | 0.576 | 8000 |
| *445.597* | 5.787 | **141.894** | 0.301 | 9000 |
| *501.995* | 6.519 | 120.51 | 0.292 | 10000 |
| *346.277* | 4.497 | 33.641 | **0.168** | 11000 |
| *351.659* | 4.567 | 30.619 | 0.308 | 12000 |
| *356.138* | 4.625 | 36.102 | 0.51 | 13000 |
| *345.502* | 4.487 | 35.596 | 0.257 | 14000 |
| ***337.908*** | **4.388** | 35.562 | 0.351 | **15000** |
| *459.406* | 6.045 | 78.891 | 0.332 | 16000 |
| *379.049* | 4.987 | 31.589 | 0.4 | 17000 |
| *423.298* | 5.57 | 78.951 | 0.489 | 18000 |
| *533.679* | 7.022 | 135.88 | 1.016 | 19000 |
| *540.838* | 7.116 | 135.965 | 0.544 | 20000 |

**Table A9**. Pixel error using RMSE scores from the Dlcrnet_ms5 training on the camera facing the nest entrance and with 75 labelled body parts across 360 frames for each iteration.

| *Sum RMSE* | *Mean RMSE* | *Highest RSME* | *Lowest RMSE* | *Iterations* |
|---|---|---|---|---|
| *445.21* | 5.708 | 65.44 | 0.696 | 5000 |
| *300.101* | 3.847 | 22.712 | 0.297 | 6000 |
| *566.174* | 7.353 | **129.785** | 0.687 | 7000 |
| *348.401* | 4.525 | 53.207 | 0.097 | 8000 |
| ***236.66*** | **3.155** | 32.456 | **0.047** | **9000** |
| *289.4* | 3.859 | 43.716 | 0.35 | 10000 |
| *456.73* | 6.01 | 101.9 | 0.07 | 11000 |
| *415.413* | 5.326 | 36.385 | 0.457 | 12000 |
| *312.394* | 4.165 | 32.936 | 0.263 | 13000 |
| *328.013* | 4.374 | 32.836 | 0.148 | 14000 |
| *300.556* | 3.955 | 27.426 | 0.385 | 15000 |
| *369.004* | 4.92 | 75.172 | 0.082 | 16000 |
| *343.0184* | 4.513 | 26.851 | 0.715 | 17000 |
| *384.741* | 5.062 | 47.108 | 0.698 | 18000 |
| *415.773* | 5.471 | 58.231 | 0.276 | 19000 |
| *314.761* | 4.254 | 30.13 | 0.391 | 20000 |

**Table A10.** Dataset of all videos used for validation with counts of bumblebees recorded by DLC compared to human observation counts.

| *Videos* | *Camera setup* | *DLC Counts* | *Human observation counts* |
|---|---|---|---|
| *P1010013* | Tubes | 15 | 17 |
| *P1010014* | Tubes | 30 | 36 |
| *P1010015* | Tubes | 27 | 37 |
| *P1010016* | Tubes | 17 | 33 |
| *P1010021* | Tubes | 21 | 48 |
| *P1010022* | Tubes | 29 | 66 |
| *P1010023* | Tubes | 28 | 37 |
| *P1200596* | Tubes | 48 | 65 |
| *P1200597* | Tubes | 55 | 71 |
| *P1200598* | Tubes | 62 | 78 |
| *P1200599* | Tubes | 54 | 86 |
| *P1200600* | Tubes | 69 | 92 |
| *P1200611* | Tubes | 52 | 74 |
| *P1200612* | Tubes | 58 | 71 |
| *P1200613* | Tubes | 52 | 63 |
| *P1010009* | Nest entrance | 19 | 56 |
| *P10100010* | Nest entrance | 18 | 60 |
| *P10100011* | Nest entrance | 9 | 79 |
| *P10100012* | Nest entrance | 26 | 106 |
| *P1010018* | Nest entrance | 7 | 28 |
| *P1010019* | Nest entrance | 16 | 30 |
| *P1010020* | Nest entrance | 10 | 26 |
| *P1200601* | Nest entrance | 10 | 11 |
| *P1200607* | Nest entrance | 16 | 28 |
| *P1200608* | Nest entrance | 25 | 49 |
| *P1200609* | Nest entrance | 24 | 33 |
| *P1200615* | Nest entrance | 15 | 29 |
| *P1200616* | Nest entrance | 18 | 44 |
| *P1200617* | Nest entrance | 11 | 42 |
| *P1200590* | Nest entrance | 19 | 65 |