ORIGINAL ARTICLE

Residual nets for understanding animal behavior

Sarthak Yadav ▪ Ankur Singh Bist

S Yadav ▪ AS Bist (Corresponding author)
KIET Group of Institutions Ghaziabad, India.
email: ankur1990bist@gmail.com

Received: July 10, 2018 ▪ Revised: September 16, 2018 ▪ Accepted: September 29, 2018

Abstract Analysis of animal behavior requires proper algorithms for the extraction of desired information from videos. Animal behavior involves various states like facial expression, body movement etc. With the advancement in hardware, deep learning has become popular for analyzing the complex and large dataset. Deep learning algorithms have proved their significance on the benchmark dataset. In this paper, we used Residual Nets for classifying three-hour video containing egg laying induced activity changes in Drosophila. We obtained 99.5% accuracy and found significant improvement in accuracy as compared to CNN (Convolutional Neural Networks). Further, it is suggested that this technique can be used for analysis of animal behavior as well as activities of other domain like object detection, speech recognition, and character recognition, among others.

Keywords: animal behavior, convolutional neural networks, residual networks

Introduction

One of the common behavior of animals is living in-group. This behavior helps in social learning, safety from predators etc. Some other behaviors related to food searching, movement, response towards external environment, mating trends, among others, attracts the attention of researchers. With the advancement of computation techniques, animal behavior can be simulated and various modeling can be done. Generally, video recordings are used to analyze animal behavior. To analyze the high volume length recording is not an easy task for researchers. Machine learning is very helpful in this scenario. With the advancement in technology, deep learning techniques are now very popular (Deng and Dong 2014, Szegedy et al 2017, Sünderhauf et al 2018).

The contribution of this work contains a modified residual net for analyzing animal behavior from eight-hour videos. The organization of the paper is as follows: This section gives complete details of recent literature of analyzing animal behavior using machine learning. Section 2 involves details about the proposed algorithm. Section 3 provides experimental work. Section 4 explains the conclusion and future work. Table1 describes a comparative analysis of past work. Many papers have been written to analyze the behavior of animals using machine learning or deep learning.

Theory

Residual networks

The advent of Deep Residual Networks (ResNets) [cite-resnet] has revolutionized Deep Learning, dominating and pushing the state-of-the-art in various applications ranging from Computer Vision (Object Detection and Localization, Semantic Segmentation) to Speech Processing (Automatic Speech Recognition).

Recent developments suggest that network depth is of paramount importance, as evidenced by results on the ImageNet dataset [cite-imagenet] in the past few years, where performance has seen a steady rise with network depth. However, simply increasing network depth by naively stacking convolutional layers is inefficient and insufficient and does not reap the benefits of increased depth due to the following problems:

1. Vanishing Gradients/Exploding Gradients: Neural Networks are trained using Backpropagation of gradients from the final layer to the first layer of the network. Vanishing/Exploding gradients is a condition that causes the gradients to vanish/explode (respectively) and thus hamper the convergence of the network. Increased depth requires gradients to backpropagate farther, thus causing convergence problems.
2. Degradation Problem: As reported in [cite-here] and verified in [cite-resnet], accuracy is saturated and then degrades rapidly with increasing network depth. This degradation is not caused by overfitting, and adding more layers to an aptly deep network only worsens the training error.

Due to the aforementioned problems, proper Network Architecture design is a crucial step for completely leveraging the approximation capabilities of Deep Neural Networks, as

doi.org/10.31893/2318-1265jabb.v7n2p97-103

evidenced by a large no. of architectures out there. [Cite-ImageNet-architectures-here].

Deep Residual Learning is one such paradigm which allows for the creation of very deep Convolutional Neural Networks (CNNs), up to 1000-layers deep [cite-here], by overcoming the aforementioned obstacles using the Residual Learning ideology. This ideology of designing neural network architectures has pushed the state of the art on a wide range of Computer Vision tasks, such as Object Detection and Localization, Segmentation, Human Activity Recognition, etc.

Residual learning

Given $h(x)$, the underlying mapping to be learned by a stack of convolution layers. In residual learning, rather than approximating $h(x)$ like a conventional one-on-one stack of layers, these stacks are explicitly designed to approximate a residual function $f(x)=h(x)-x$. Thus the original function becomes $h(x)=f(x)+x$.

Identity mapping

Residual learning is adapted to every few stacked layers, which are denoted as building blocks (Fig. 1). Formally, each building block is defined as:

$$y=F(x,Wi)+x$$

where x and y are the input and output vectors of the building block in question and the function $F(x,Wi)$ represents the mapping to be learned. Therefore, for the example in Figure 1, $F(x,Wi)=W_2\sigma(W_1x)$, in which σ denotes the non-linear activation function, most commonly ReLU [cite-relu]. The operation $F+x$ is performed by a shortcut connection in an element-wise fashion. The non-linearity is re-applied after the addition (i.e. $\sigma(y)$).

The dimensions of $F(x)$ and x must be equal for this simplistic identity mapping to work. In case they are not equal, a linear projection of W_S is applied to x such that the dimensions now match.

$$y=F(x, W_i)+ W_S\, x$$

Adam optimizer

Adaptive Moment Estimation, also called Adam, is a first-order gradient-based optimization algorithm with adaptive learning rates for each parameter. Adam adjusts the learning rate of every parameter individually at every gradient update on the basis of the past gradients of the parameter. Following is a brief overview of gradient update rule in Adam. Adam stores the following information:

1. Exponentially decaying average of past squared gradients at iteration t, denoted by $\mathbf{v_t}$
2. Exponentially decaying average of the past gradients at iteration t, denoted by m_t

$$m_t=\beta_1 m_{t-1}+(1-\beta_1)g_t$$

$$v_t=\beta_2 v_{t-1}+(1-\beta_2)g_t^2$$

where g_t is the gradient at t, and β_1 and β_2 are constants. m_t and v_t are initialized as vectors of zeros, therefore we need to apply bias correction:

$$m_t^{'}=\frac{m_t}{(1-\beta_1^t)}$$

$$v_t^{'}=\frac{v_t}{(1-\beta_2^t)}$$

Finally, the gradient update rule for parameter Θ can be defined as:

$$\Phi_{t+1}=\Phi_t-\frac{\eta}{\sqrt{v_t^{'}+\epsilon}}m_t^{t}$$

where η is the initial learning rate. The suggested default values are β_1=0.9, β_2=0.999 and $\varepsilon=10^{-8}$.

Due to its adaptive learning rate property, Adam resolves the need for complicated learning rate schedules and eases training of deep networks.

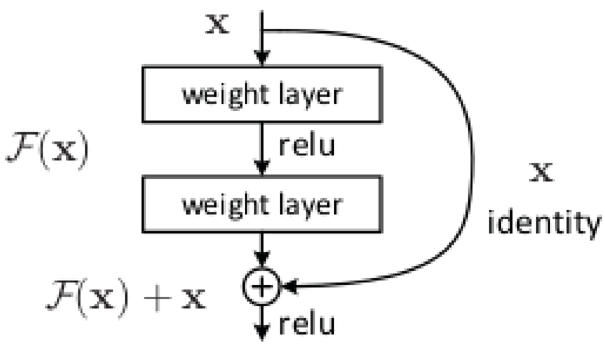

Figure 1 Resnet building block.

doi.org/10.31893/2318-1265jabb.v7n2p97-103

Table 1 Related works in chronological order.

S.No.	Year	Title	Technique and Results
1.	Pons et al 2017	Accessing machine learning classifiers for the detection of animals' behaviour using depth based tracking	Authors analyzed posture and orientation of cats and applied depth based trackers and obtained promising results.
2.	Dalziel et al 2008	Fitting probability distributions to animal movement trajectories: Using artificial neural networks to link distance, resource and memory	Authors used artificial neural network on five elk to calculate the movemnt prabability kernels of three behavioral processes of landscape.
3.	Nadimi et al 2008	ZigBee based wireless sensor networks for classifying the behaviour of a herd of animals using classification trees	Authors selected pitch angle of the neck and velocity, they applied fuzzy logic classifiers and found improved results.
4.	Hokkanen et al 2011	Predicting sleep and lying time of calves with a support vector machine classifier using accelerometer data	Authors took data from 10 dairy calves and perform wavelet analysis, support vector machine and found 82+2% occurrence of sleep.
5.	Cheng et al 2012	A comparative study in birds: call-type-independent species and individual recognition using four machine learning methods and two acoustic features	Authors used support vector machine using two acoustic features for individual classification activity of three passerine.
6.	Bidder et al 2014	Love Thy Neighbour: Automatic animal behavioural classification of acceleration data using the K-Nearest Neighbour Algorithm	Authors used K-Nearest Neighbour (KNN) for analyzing animal position and found this primitive technique suitable for the same situation.
7.	Dutta et al 2015	Dynamic cattle behavioural classification using suoervised ensemble classifier	Authors used bagging ensemble with tree learner and obtained 96% accuracy for cattle behaviour classification
8.	Nilsson et al 2015	Development of automatic surveillance of animal behaviour and welfare using image analysis and machine learned segmentation technique	Authors used computer vision technique that uses learning based segmentation on extracted features. Results shows the effectiveness of learning based methods as compared to grey scale techniques.
9.	Shahriar et al 2015	Heat event detection in dairy cows with collar sensors: An unsupervised machine learning approach	Authors extracted features in the form of standard deviation, energy, amplitude and fast fourier transform. K-Means algorithm is used on the dataset and activity index level was obtained.
10.	Smith et al 2015	Bag of class posteriors, a new multivariate time series classifier applied to animal behaviour identification	Authors proposed novel multi-scale time series classifier for large animal behaviour dataset and found it more suitable asa compared to bag of features model.
11.	Stern et al 2015	Analyzing animal behaviour via classifying each video frame using convolutional neural networks	Authors used convolutional neural network on features extracted from video frame and have 0.072% error rate only.
12.	Ladds et al 2016	Seeing it all: Evaluating supervised machine learning methods for classification of diverse otariid behaviours	Authors extracted 26 behaviours from video and found that SVM with polynomial kernel has high accuracy as compared to random forest and GBM (Gradient Boosting Metho)
13.	Leos-Barajas et al 2016	Analysis of animal accelerometer data using hidden markov models	Authors analyzed acceleration data and apply hidden markov model and draw useful inferences
14.	Rahman et al 2016	A comparison of autoencoder and statistical features for cattle behaviour classification	Authors used stacked autoencoder for cattle behaviour classification and found it better as compared to statistical features, taken from past analysis of behaviour motion.
15.	Norouzzadeh et al 2018	Automatically identifying, counting, and describing wild animals in camera-trap images with deep learning	Authors took 3.2 million images from 48 species and applied deep convolutional neural network and obtained 92% accuracy.
16.	Villa et al 2017	Towards automatic wild monitoring: Identification of animal species in camera trap images using very deep convolutional neural network	Authors used SSe dataset and used very deep convolutional neural network for unbalance data algorithm and found 35.4% top1 and 60.4% top5 accuracy.

17.	Valletta et al 2017	Applications of machine learning in animal behaviour studies	Authors presented the case studies using mahine learning on Pheasant Eggs, Jackdaw Associations & Wildebeest.
18.	Morota et al 2018	Big data analytics and precision animal agriculture symposium: Machine learning and data mining advance predictive big data analysis in precision animal agriculture	Authors identified the data processing and inference generation problem of animal sciences and showed how machine learning and data mining approaches can be helpful in concerned field.
19.	Wild et al 2018	Automatic localization and decoding of honeybee markers using deep convolutional neural networks	Authors proposed deep convolutional neural networks for localization and decoding of custom binary markers and found this technique comparable with recent papers in literature.
20.	Kumar et al 2018	Deep learning framework for recognition of cattle using muzzle point image pattern	Authors designed muzzle point image database using deep learning approach and found 98.99% accuracy.
21.	Doucette et al 2018	Machine learning based classification of deep brain stimulation outcomes in a rat model of binge eating using ventral striatal oscillations	From NAC (Nucleus Accumbens), LFPs (Local field potentials) are extracted as features and lasso is applied as prediction algorithm.
22.	Gutha et al 2018	Effect of abiotic and biotic stress factors analysis using machine learning methods in zebrafish	Authors used recursive support vector machine using differentially expressed genes and found 100% accuracy. Other than this random forest perform better in eight different stress conditions.
23.	Xia et al 2018	Aquantic toxic analysis by monitoring fish behaviour using computer vision: A recent progress	Authors tracked group of individuals in 2D and 3D space by taking video using machine learning techniques and introduced the advantage of deep learning for same problem.

Batch normalization

Input normalization is used extensively to speed up training of neural networks by bringing all input values in the same range. In simple words, Batch Normalization is input normalization for hidden layers of a deep neural network. Batch Normalization is a technique that increases training speed by reducing Internal Covariate Shift. Internal Covariate shift is the change in the distribution of network activations due to the change in network parameters during training. [cite-batchnorm] Batch Normalization, when applied before a hidden layer, reduces internal covariate shift by normalizing inputs according to the mean and standard variation of the output activations of the previous layer.

For a layer with d-dimensional input $x= [x^1, x^2 \ldots x^d]$, each dimension is normalized as follows:

$$x'^k = \frac{x^k - \mathrm{E}[x^k]}{\sqrt{Var[x^k]}}$$

where the expectation E and variance V are computed over the training data set. Batch Normalization adds two trainable parameters (for each layer), γ and β, which scale and shift the normalized value.

Let a mini-batch $B=x_{1\ldots m}$, where x_i is the i^{th} input and m is the batch size. Then the output, y_i can be defined as:

$$y_i = \mathrm{B}N_{\gamma,\beta^{(x_i)}}$$

$$y_i = \gamma\hat{x}_i + \beta$$

For the sake of brevity, training procedures for γ and β are omitted.

Dropout

Given the huge number of parameters in a deep CNN, (eg. VGG 16 has 140 Million parameters), these networks are tedious to train and prone to overfitting, resulting in poor generalization performance.

Dropout is a technique for aiding with overfitting (regularization) as well as approximately combining exponentially many networks efficiently. Applying dropout to a neural network can be thought of as "sampling" a thinned network from it, the thinned network comprising of all units, which survived dropout. Dropout works differently at training time and inference time.

1.Training Time: In simplest case, each hidden unit is retained with a probability of p, or dropped out (its outputs are zeroed) with a probability $1-p$, independent of other units. The optimal probability of retention is usually closer to 1 rather than 0.5.

2.Inference Time: At test time, the neural network is used without dropout. The weights of this network are scaled down version of the trained nets. This is governed by the following simple rule: "if a unit is retained with a probability p during training, the outgoing weights of that unit are multiplied by p at test time". This way, 2^n networks with shared weights can be combined into a single neural network at test time.

Materials and Methods

The following subsections describe the proposed approach: Model Architecture

We propose a custom model architecture that is inspired by Residual Networks (ResNets). For clarity and ease of representation, we have divided our network into 2 constituent blocks: Basic Block with Identity Skip Connections (Figure 2) and Basic Block with Down sampling skip connections (Figure 3).

Basic block with down sampling skip connections allow us to accommodate to reductions in feature map sizes, and are used sparingly. All convolution operations comprise of 3x3 filters, with *padding* = 1 and *stride* = 1, unless specified otherwise.

The proposed network comprises of a total of 14 weighted layers (13 Convolutional Layers and 1 Linear layer). Due to much smaller input size (92x92), the first layer of our network comprises of 3x3 Convolutions at a stride of 1, as opposed to the originally proposed ResNets where the first layer utilized an arrangement which had a much larger field of view (7x7, stride=2). With 2.8 Million parameters, the proposed network is approximately 4x smaller than the smallest ResNet configuration (Resnet 18).

We used Dropout with a drop probability of 40% before the final Linear Layer in order to reduce overfitting. Another key difference between the proposed architecture and the original architecture is that throughout the network, we apply Batch Normalization after the non-linearity (ReLU).

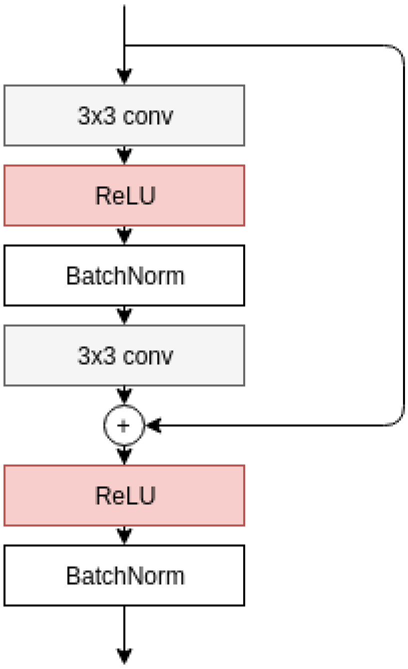

Figure 2 Basic block without down sampling.

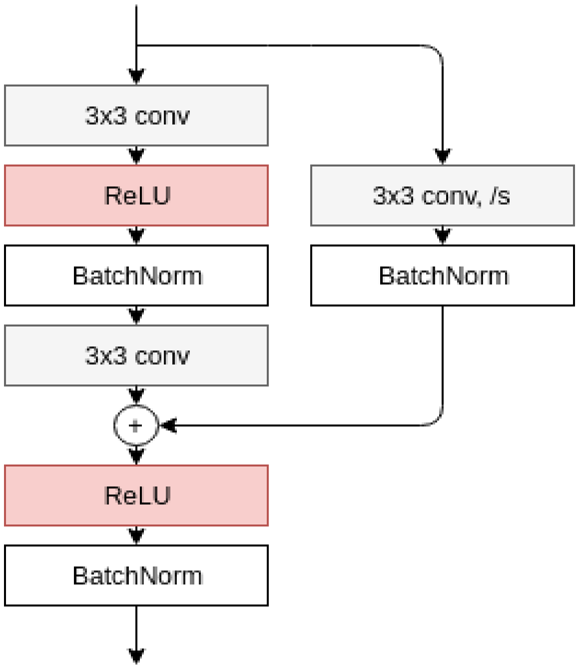

Figure 3 Basic block with down sampling.

Pre-processing and data augmentation

We utilize per image normalization (removing mean and dividing by the standard deviation) of pixel values for the input grayscale images.

Further, to aid with over-fitting, we use the following transforms at training time:

1. Random horizontal flipping of images
2. Random rotation in the range of [−10,10] degrees.
3. Random horizontal scaling and Random horizontal and vertical translations.

As opposed to [originalpaper], we do not utilize any kind of test time augmentation.

Loss function

The network is trained using Softmax Loss, which can be formulated as follows:

$$L_s = -\sum_{i=1}^{n} \log \frac{e^{W_{y_i}^T x_i + b_{y_i}}}{\sum_{j=1}^{n} e^{W_j^T x_i + b_{y_j}}} \quad (1)$$

where n denotes the number of classes. For our case, n=2.

Results and Discussion

We use the same training and test split as given in Rahman et al (2016). For hyper-parameter selection, we utilized four fold Cross Validation on the training set. We used the Adam optimization algorithm, with the default parameters, for training the network with a batch size of 64. The proposed approach achieved a Test accuracy of 99.5% using a single Network, as opposed to Rahman et al (2016), where they achieved a classification accuracy of 92.8% using a 30-model average, which is a drastic increase in performance. After experiments, we found the following results: The model architecture, AUC ROC Curve and the confusion Matrix (Figures 4, 5 and 6, respectively).

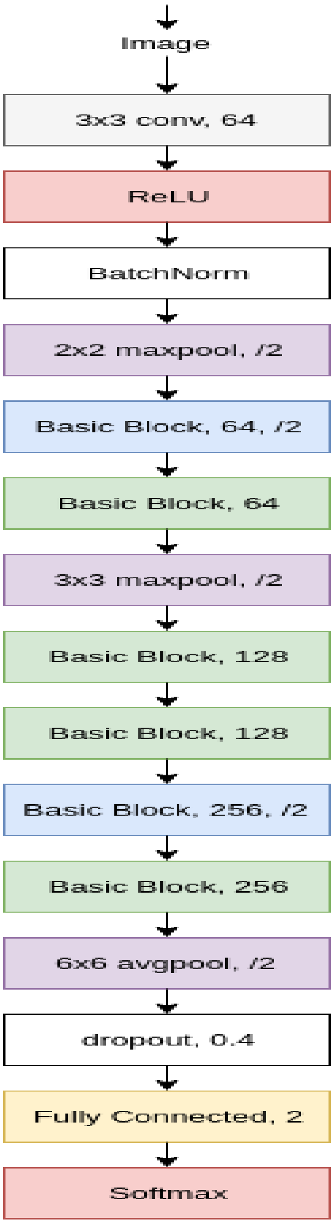

Figure 4 Model architecture.

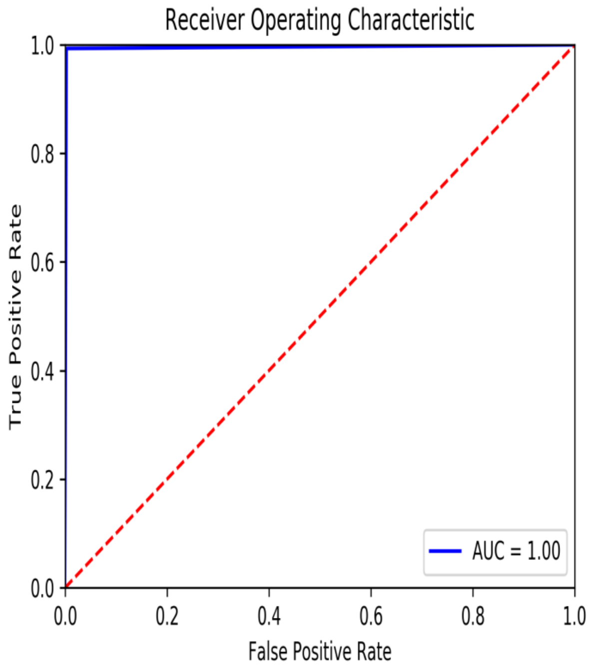

Figure 5 AUC ROC curve.

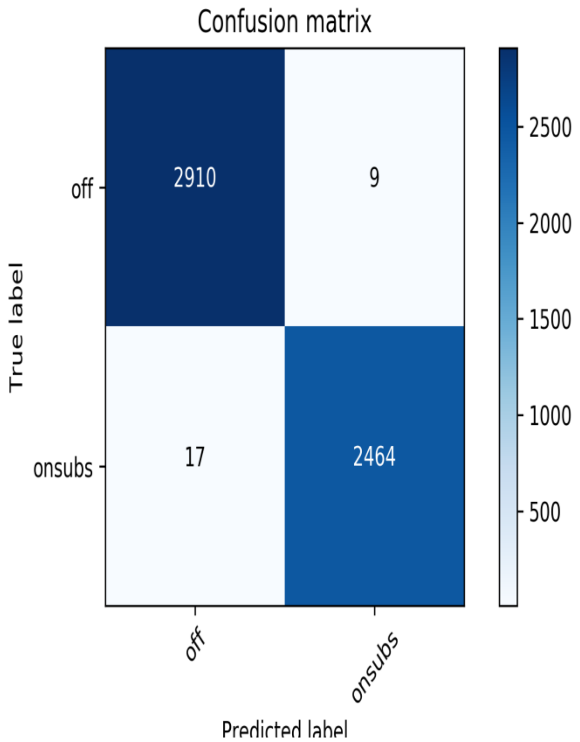

Figure 6 Confusion matrix.

Conclusions

Advancements in deep learning attracted ours towards residuais networks, initially developed by Microsoft researchers. That is why in this paper we used Residual Nets for classifying three-hour video containing egg laying induced activity changes in Drosophila and compared it with [14]. Comparison of our proposed work with past reflects the significance of our work. In the future, we will take a more advanced version of deep learning algorithms for solving other problems related to animal behavior.

References

Deng L, Dong Y (2014) Deep learning: methods and applications. Foundations and Trends® in Signal Processing 7: 197-387.

Szegedy C, Ioffe S, Vanhoucke V, Alemi A (2017) Inception-v4, inception-resnet and the impact of residual connections on learning. AAAI 4:15-30.

Sünderhauf N, Brock O, Scheirer W, Hadsell R, Fox D, Leitner J, Upcroft B, Abbeel P, Burgard W, Milford M, Corke P (2018) The limits and potentials of deep learning for robotics.The International Journal of Robotics Research 37:405-420.

Patricia P, Jaen J, Catala A (2017) Assessing machine learning classifiers for the detection of animals' behavior using depth-based tracking. Expert Systems with Applications 86:235-246.

Dalziel, Benjamin D, Morales JM, Fryxell JM (2008) Fitting probability distributions to animal movement trajectories: using artificial neural networks to link distance, resources, and memory. The American Naturalist 2:248-258.

Nadimi, Shahrak E, Søgaard HT, Bak T (2008) ZigBee-based wireless sensor networks for classifying the behaviour of a herd of animals using classification trees. Biosystems engineering 100:167-176.

Hokkanen, Helena A (2011) Predicting sleep and lying time of calves with a support vector machine classifier using accelerometer data. Applied Animal Behaviour Science 134:10-15.

Jinkui C (2012) A comparative study in birds: call-type-independent species and individual recognition using four machine-learning methods and two acoustic features. Bioacoustics 21:157-171.

Bidder, Owen R (2014) Love thy neighbour: automatic animal behavioural classification of acceleration data using the k-nearest neighbour algorithm. PloS one 10:15-25.

Ritaban D (2015) Dynamic cattle behavioural classification using supervised ensemble classifiers. Computers and Electronics in Agriculture 111:18-28.

Mikael N (2015) Development of automatic surveillance of animal behaviour and welfare using image analysis and machine learned segmentation technique. Animal 11:1859-1865.

MdSumon S (2015) Heat event detection in dairy cows with collar sensors: An unsupervised machine learning approach. Sensors 10:15-21.

Daniel S (2015) Bag of class posteriors, a new multivariate time series classifier applied to animal behaviour identification. Expert Systems with Applications 42:3774-3784.

Ulrich S, He R, Yang CH (2015) Analyzing animal behavior via classifying each video frame using convolutional neural networks. Scientific reports 5:14351.

Ladds, Monique A (2016) Seeing it all: evaluating supervised machine learning methods for the classification of diverse otariidbehaviours. PloS one 11:18-28.

Barajas L, Vianey (2017) "Analysis of animal accelerometer data using hidden Markov models. Methods in Ecology and Evolution 8:161-173.

Ashfaqur R (2016) A comparison of autoencoder and statistical features for cattle behaviour classification. Neural Networks (IJCNN) International Joint Conference on IEEE.

Norouzzadeh MS, Nguyen A, Kosmala M, Swanson A, Packer C, Clune J (2018) Automatically identifying, counting, and describing wild animals in camera-trap images with deep learning. PNAS 115:E5716-E5725.

Villa AG, Salazar A, Vargas F (2017) Towards automatic wild animal monitoring: identification of animal species in camera-trap images using very deep convolutional neural networks. Ecological Informatics 41:24-32.

Valletta, JJ (2017) Applications of machine learning in animal behaviour studies. Animal Behaviour 124:203-220.

Morota G (2018) Big data analytics and precision animal agriculture symposium: Machine learning and data mining advance predictive big data analysis in precision animal agriculture. Journal of Animal Science 96:1540-1550.

doi.org/10.31893/2318-1265jabb.v7n2p97-103