Soft Computing
https://doi.org/10.1007/s00500-021-06493-8

APPLICATION OF SOFT COMPUTING

Deep vision-based surveillance system to prevent train–elephant collisions

Surbhi Gupta[1] · Neeraj Mohan[2] · Padmalaya Nayak[3] · Krishna Chythanya Nagaraju[3] · Madhavi Karanam[3]

Accepted: 23 October 2021

Abstract

Animal conservation is imperative, and technology can certainly assist in different ways. The extinction of endangered species like tigers and elephants has boosted the necessity for such efforts. Human–elephant collision (HEC) has been an active area of research for years. Apart from deforestation, the roads and rail tracks laid down through forest areas intervene a lot in wildlife. Collisions and tragedies are every day, especially in green belts in India and other Asian countries. Therefore, it is crucial to develop vision-based, automated, warning-generating systems to identify the animal/elephant near-site. In the proposed work, different deep learning-based models are proposed to identify elephants in image/video. Several convolutional neural network (CNN)-based models and three transfer learning (TL)-based models, i.e., ResNet50, MobileNet, Inception V3, have been experimented with and tuned for elephant detection. All the models are tested on a synthesized dataset having about 4200 images built using two public datasets, i.e., ELPephant and RailSem19. Two accurate CNN and transfer learning-based models are presented in detail. These highly accurate and precise models can alarm the trains and generate warning signals on site. The proposed CNN and inception network demonstrated high accuracy of 99.53% and 99.91%, respectively, and are remarkable in identifying elephants and hence preventing HEC. The same model can be trained for other animals for their preservation in similar scenarios.

Keywords Human–elephant collision · Rail track monitoring · Deep vision · Data augmentation · Transfer learning

1 Introduction

Animal mortality is becoming a critical concern worldwide as it is disturbing ecological balance, and many animal species are being endangered. The International Union for Conversation of Nature (IUCN) has already endangered the Indian elephant. Particularly in India, human life is intrinsically entangled with the giant animal elephant. Whether it is culture, mythology, or the Hindu custom, elephant is considered a sacred animal and the symbol of intellectual strength. In Hinduism, every auspicious work usually starts with a prayer to the elephant-faced Ganesh.

Nevertheless, the life of elephants is full of struggle in the present scenario. Due to much human intervention in their habitat, it is an endangered species now. The reason could be the costly ivory tusk of the mammal, the deforestation by the greedy human beings, or collision with humans; at the receiving end are the elephants (Langbein 2011; Morse et al. 2014). Poaching of elephants is a common incident, and the government is taking serious action to prevent this. Wild-life sanctuaries and parks have

✉ Surbhi Gupta
surbhi@pau.edu

Neeraj Mohan
erneerajmohan@gmail.com

Padmalaya Nayak
drpadmalaya2010@gmail.com

Krishna Chythanya Nagaraju
kcn_be@rediffmail.com

Madhavi Karanam
bmadhaviranjan@yahoo.com

1 Department of Electrical Engineering and IT, COAET, Punjab Agricultural University, Ludhiana, India

2 Department of CSE, IK Gujral Punjab Technical University, Mohali, India

3 Department of CSE, Gokaraju Rangaraju Institute of Engineering and Technology, Hyderabad, India

Published online: 12 November 2021

been made to preserve their habitat. But despite these efforts, the collisions and mishappening due to train accidents are causing a great loss. In Asian countries like India, trains are the cheapest and mass transport mechanism. Several rail tracks exist in/near the green zone or forest areas for connectivity. Small, hasty animals are alert enough to act upon whenever they feel and realize the danger of trains, while bulky animals like an elephant could not manage to save themselves and hence lose their life. The human–elephant collision has become a significant research point owing to a declining number of Elephants. Various alert systems and barriers have been placed at binding sites but still take a significant toll on elephants. Many researchers have worked on the impact of railway tracks being passed through the forest area. Others studied the statistics of elephants killed by train collisions over the past few years (Chythanya et al. 2020) and measures taken by the Govt. of India in handling the issue meticulously.

Much effort has been put into the identification and tracking of animals using computer vision-based techniques. Three main approaches are widely used in the literature to study these problems. The first category of research is based on aerial images captured from unmanned aerial vehicles (UAVs) (Kellenberger et al. 2017, 2018a, b, 2019; Rey et al. 2017). However, the challenge is that difference between aerial and ground imagery is likely to make identification difficult. The second category of research corresponds to motion-triggered cameras to trap animal images (Chen et al. 2014; Ren et al. 2015; Redmon and Farhadi 2017; Schneider et al. 2019; Beery et al. 2018). Methods based on deep learning (Chen et al. 2014), Faster RCNN (Ren et al. 2015), and YOLOv2 (Redmon and Farhadi 2017) are used for video segmentation to extract the animal movement from the still background. However, these methods face difficulties in generalizing to new environments. The third category involves real-time videos captured using ground-level cameras (Mönck et al. 2018; Patman et al. 2018). These types of surveillance and safety applications are much rarer in the literature, and hence this motivated us for the present work.

Image processing powered with machine learning and deep neural networks has already given promising results in similar real-time applications. Many attractive solutions for plant disease identification (Gadekallu et al. 2020) and plant species identification (Kumar et al. 2019a) have been explored. Deep learning solutions can be combined with optimization algorithms to solve complex problems like hand gesture classification (Gadekallu et al. 2021). Recently, a multilayer feedforward neural network-based approach is proposed for human–robot collision detection (Sharkawy et al. 2020). Similarly, a collision-avoidance system is proposed for a biomimetic autonomous underwater vehicle using an artificial neural network (Praczyk 2020). The major challenge is developing a fast, low-cost, and accurate solution that can be achieved using deep learning models. Hence, the use of deep learning for the detection of human–elephant collision needs to be explored. Thus, the main contributions of this paper are:

1. To review the existing efforts for human–elephant collision prevention
2. To propose an accurate CNN architecture for elephant detection on the rail track
3. To explore the usage of transfer learning (TL)-based models for efficient detection of the elephant in different positions on rail track so that an efficient alert system can be implemented
4. To experiment with three TL models, i.e., ResNet50, MobileNet, and Inception V3, for precise identification of elephant on track
5. A complete model for alarm generation to avoid HEC and hence animal conservation.

2 Related literature

Many researchers have worked on identifying animals in images/videos that help prevent human and animal collisions. Animal detection algorithms using machine learning and deep learning are emerging. Koik and Ibrahim (2012) and Tanwar et al. (2017) presented a literature survey on animal detection methods using digital images. We have conducted a survey on the existing contribution on HEC, and the summary is shown in Table 1.

Several effective approaches for animal detection and tracking have been proposed (Tweed and Calway 2002; Ramanan and Forsyth 2003; Hannuna et al. 2005; Burghardt and Ćalić 2006; Farah et al. 2011; Mammeri et al. 2014; Sharma and Shah 2016; Norouzzadeh et al. 2018; Raja et al. 2018; Devost et al. 2019; Zotin and Proskurin 2019; Backs et al. 2017; Bíl et al. 2019; Jayakumar et al. 2020). Sharma and Shah (2016) proposed real-time animal detection and collision avoidance system using a computer vision technique. Norouzzadeh and Nguyen (2018) performed identifying and counting wild animals in camera-trap images with deep learning. The prevention of wild animals from accidents using image detection and edge algorithm was studied by Raja et al. (2018). Devost et al. (2019) proposed a new automated tool for animal detection in camera trap images. Zotin and Proskurin (2019) performed animal detection using a series of images under complex shooting conditions.

Backs et al. (2017) proposed low-cost electronic-based devices to avoid train collision with animals on track. They performed testing on Bayy National Park and Yoho

Table 1 Summary of contribution for prevention of human–animal/elephant collision

Approaches applied for	Author	Proposed method
Animal detection and tracking	Tweed and Calway (2002)	Movement track model using condensation particle filtering
	Ramanan and Forsyth (2003)	Temporal coherency-based model for animal detection
	Hannuna et al. (2005)	Animal gait identification model
	Burghardt and Ćalić (2006)	Haar-like feature extraction with Adaboost classifier-based model
	Farah et al. (2011)	A robust method to track animals and determine their motion pattern
	Mammeri et al. (2014)	Two-step classification system using LBP-Adaboost followed by HOG-SVM Classifier
	Sharma and Shah (2016)	Animal detection and collision avoidance system using Computer Vision
	Norouzzadeh and Nguyen (2018)	Automatically identifying, counting, and describing wild animals in camera-trap images with Deep Learning
	Raja et al. (2018)	Image identification using an edge detection algorithm
	Devost et al. (2019)	Automated tool for animal detection in camera trap images
	Zotin and Proskurin (2019)	Animal detection using a series of images under complex shooting conditions
	Backs et al. (2017)	Hebb's Law of a learning-based model
	Bill et al. (2019)	Kernel density estimation-based model
	Jayakumar et al. (2020)	Animal detection using a deep learning algorithm
Elephant detection and tracking	Venkataraman et al. (2005)	Satellite image-based elephant position of African elephant
	Ardovini et al. (2008)	Ear shape-based model for an elephant identification system
	Vermeulen et al. (2013)	Unmanned aerial vehicles at the height of 100mts for tracking
	Zeppelzauer (2013)	Automated detection of elephants in wildlife video
	Sugumar and Jayaparvathy, (2014)	Image feature extraction and similarity matching model based on Euclidian and Manhattan distance
	Zeppelzauer et al. (2015)	Color model-based elephant detection
	Shukla et al. (2017)	Computer vision framework for detecting and preventing human–elephant collisions
	Mandal et al. (2018)	Unsupervised Artificial Neural Network model using Geophone Sensors to detect elephants
	Marais et al. (2018)	Automated elephant detection and classification from aerial infrared and colour images using deep learning
	Dhanaraj and Sangaiah et al. (2018)	Elephant detection using boundary sense deep learning (BSDL) architecture
	Kumar et al. (2019b)	Detection of wild elephants using image processing on Raspberry Pi3
	Ravikumar et al. (2020)	Transfer learning-based MobileNet model for elephant detection

National Park, where grizzly bears, Black bears, Wolves, and Moose were found to be killed in large numbers. The first method uses two paired devices placed distantly. The first device detected the passing train and sent information to the paired warning device placed at a high striking rate position. Another method includes predicting train arrival time at a distance hypothetically considered as 200 m and activates an integrated warning mechanism at the desired time. Random Forest classification models were used with tenfold 10-repeat cross-validation, and an 80% detection rate was claimed. Bill et al. (2019) categorized HEC as occurrences in the cluster and outside the cluster. The kernel density estimation (KDE) method was extended with a Monte Carlo simulation called KDE + to identify more significant clusters. They achieved a 95% confidence interval, and road width, shrubs, and habitat type were observed as the most significant variables. Jayakumar et al. (2020) proposed animal detection using a deep learning algorithm.

Other authors made efforts for the detection and tracking of elephants specifically (Venkataraman et al. 2005; Ardovini et al. 2008; Vermeulen et al. 2013; Zeppelzauer 2013; Sugumar and Jayaparvathy 2014; Zeppelzauer and Stoeger 2015; Shukla et al. 2017; Mandal and Bhutia 2018; Marais 2018; Dhanaraj and Kumar Sangaiah 2018; Kumar et al. 2019b; Ravikumar et al. 2020). Initially, feature extraction was used to detect elephants in frames. Sugumar and Jayaparvathy (2014) used images captured with a camera mounted on towers or trees and sent to a distant base station through an RF network. Image feature extraction and similarity matching based on Euclidian and Manhattan distance was done with the help of multilevel wavelet coefficients obtained Haar wavelet decomposition of the image. K-means clustering is used to cluster images, and F-Norm theory was employed to identify the need for sending warning SMS. The researchers used an optimized distance measure that gave an 18.5% better performance than other measures. Zeppelzauer (2015) implemented the detection of elephants in wildlife videos. Shukla et al. (2017) discussed a computer vision framework for detecting and preventing human–elephant collisions. The author mentioned that vision-based approaches are more reliable as compared to other techniques. The researchers used deep conventional neural networks to recognize elephants and elephant faces and track the movement of an animal with a warning message in case of conflict. The researchers made use of the transfer learning model Alex-Net with 4096 × 1 dimension. This representation is fed to support vector machines for binary classification. The work was carried out using a dataset of twelve videos comprising 500 images of elephants. The researchers claim to achieve high precision of 98.6%. The particle filter method with a color change from Blue to Red is used to detect human–elephant collisions.

Mandal et al. (2018) proposed an unsupervised artificial neural network model using Geophone Sensors to detect elephants near the railway tracks, which is supposed to activate Bee sound or virtual cracker fire sound to scare out the Elephant. Marais (2018) proposed automated elephant detection and classification using aerial infrared and color images using deep learning. Similarly, Dhanaraj and Sangaiah (2018) did elephant detection use boundary sense deep learning (BSDL) architecture. Kumar et al. (2019b) performed the detection of wild elephants using IOT-based solution. Finally, Ravikumar et al. (2020) proposed MobileNet architecture for successful elephant detection.

The problem of elephant mortality on railway tracks is persistent, and certainly environmental activists trying to throw light on this problem through social media may sometimes face cyberbullying by a few vested interest groups including poachers, smugglers, road contractors, railway contractors or politicians, etc. Cyberbullying using text in social media has attracted a few research works in the recent past with advances in deep learning. The authors of Iwendi et al. (2020) have researched identifying cyberbullying in social media; they worked on the Kaggle data set to find insults based on messages in social media considering single class classification as 0—insulting or 1—neutral. Four different deep learning models including recurrent neural networks, gated recurrent neural networks, gated recurrent units, long short-term memory, bidirectional long short memory.

Although many methods (Tweed and Calway 2002; Ramanan and Forsyth 2003; Hannuna et al. 2005; Burghardt and Ćalić 2006; Farah et al. 2011; Mammeri et al. 2014; Sharma and Shah 2016; Norouzzadeh et al. 2018; Raja et al. 2018; Devost et al. 2019; Zotin and Proskurin 2019; Backs et al. 2017; Bíl et al. 2019; Jayakumar et al. 2020; Venkataraman et al. 2005; Ardovini et al. 2008; Vermeulen et al. 2013; Zeppelzauer 2013; Sugumar and Jayaparvathy 2014; Zeppelzauer and Stoeger 2015; Shukla et al. 2017; Mandal and Bhutia 2018; Marais 2018; Dhanaraj and Kumar Sangaiah 2018; Kumar et al. 2019b; Ravikumar et al. 2020) were proposed for animal and elephant detection, some open issues need to be addressed appropriately:

- One most important issue is the lack of an appropriate dataset for experimentation. We aimed at creating a synthesized dataset for HEC. The lack of training data was addressed by collecting and labeling a sufficiently large number of images of the target animals.
- Another significant challenge is the need for a fast and computationally effective detector to process streams and communicate in real-time using low-power machines. This paper aims to employ CNN and transfer learning to efficiently detect elephants in different rail tracks to implement the low-cost solution.

3 Dataset used

As the experiment aims to identify elephants on or near the rail track, we have searched for various images and datasets which serve the purpose. Naude and Joubert (2019) have presented a new public benchmark for aerial elephant detection. Many such datasets are publicly available for experimentation on animal detection, but no ideal dataset for animals on rail track is available as it is practically challenging to capture natural images of an elephant on or near the track. Some random images are available, however. Due to this limitation, we have decided to use a dataset of elephants and a dataset with rail track images and create our own synthesized dataset of about 4200 images. Two public datasets are used in this current research, i.e., ELPephant (Korschens and Denzler 2019) and RailSem19 (Zendel et al. 2019).

ELPephant dataset is a massive dataset of 2078 images with the image of single and multiple elephants. Moreover, this dataset includes different classes of elephants with 3–22 images for each class. It includes different features and different poses of elephants. Left, right, and front profiles are taken. Sitting elephants are also featured in some images. This dataset was explicitly designed for the class detection of the elephants.

As the experiment aims to identify the elephants at the track, we need images with clear rails tracks. Therefore, we have used the dataset RailSem19 as it contains many rail track images with different locations and angles. This dataset is vast as it consists of 8500 rail images from 38 countries captured in varying weather and light conditions.

Apart from the dataset mentioned above, we needed more realistic images to test the proposed model. Therefore, we used different keywords like “Elephant on the rail track,” “Animals on rails track,” etc., to search for authentic images. These images are used for training and testing purposes. Table 2 shows the distribution of the number of images used for experimentation. Approximately 20% of images are considered for testing purposes from Elephant and RailSem19 datasets. The remaining 80% are further split into training and validation sets. This split is done randomly to avoid any bias in the experiment.

Moreover, the variety in images ensured a high variance in images during the training phase. Figure 1 shows the sample images taken for experimentation. Row 1 and row 2 have the sample images from the ELPephant dataset and the RailSem19 dataset. Row 3 and 4 show random images based on google search with different keywords “animals on rail track” and “elephant on the rail track,” respectively.

4 Methodology used

Deep learning and transfer learning architectures are explored for HEC avoidance experimentation and are discussed below.

4.1 CNN architecture

CNN stands for convolutional neural networks, aiming to extract features from images to help identify the objects present in the image. These are a type of neural network with a mix of convolutional, pooling, and fully connected layers. Traditionally, in machine learning, a subject expert is supposed to handpick the features used to identify and classify objects in an image/video. With the usage of CNN architecture, feature extraction has become automatic. The convolutional layers in CNN employ a different type of filter on images. These filters can extract useful features from the images, such as color information, edge detection, etc. Multiple convolutional layers cause complex features to be extracted from the images. Then, these extracted features are reduced using pooling layers so that only significant features can pass onto the subsequent layers. This helps in reducing the computational complexity of the architecture.

At last, the reduced feature set is passed to fully connected layers. Finally, the last layer is used to classify the results in binary or multiple classes. Convolutional and pooling layers serve as feature extraction, while a fully connected layer is used for final classification.

Various CNN architectures are possible for feature extraction and classification. If changed, the number and type of layers will cause a new architecture with a different feature set and outcome. Apart from that, many hyperparameters are available to fine-tune the architecture. Different convolution layers with different parameters such as filter size, activation function, and image size can be explored for feature extraction. For the pooling layer, stride can be selected as per requirement. For dense layers, activation functions such as “Relu” or “leaky-relu” can be chosen for fully connected layers.

Further, for final classification using a dense layer, activation functions like “sigmoid” or “softmax” can be used. The sigmoid function may be used for binary, and the Softmax function may be used for multi-class classification. Further, learning rate, dropout, and regularization can be used for tuning the architecture.

4.2 Transfer learning

Transfer learning is an AI strategy where a model produced for an application is reused as the beginning stage for a model on a subsequent application. It is a well-known methodology in deep learning that pre-prepared models could be the beginning point for the next model. These pre-trained models (trained using large datasets) are saved with the weights after their training. The same model is then trained on a new dataset with some changes in top layers. Then the final model is tested for classification or other problems. The concept of transfer learning is visualized in Fig. 2.

Table 2 Total number of images considered for experimentation

Dataset	No. of images	Train-validation (80–20% internally)	Test
ELPephant	2078	1573	505
RailSem19	2227	1676	551
Elephant on track	60	52	8
Other animals on track	32	–	32

Fig. 1 Sample images used for training and testing. Row 1: images from Elephant dataset; Row 2: Images from RailSem19 dataset; Row 3: Images from the Internet with the keyword "Animals on Rail track"; Row 4: Images from the Internet with the keyword "Elephants on Rail track"

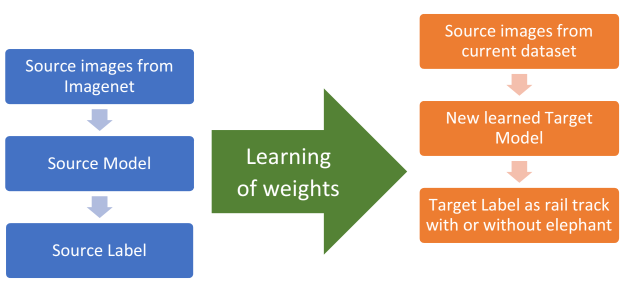

Fig. 2 Concept of transfer learning

There are three main steps in utilizing a pre-trained model. First, a pre-prepared source model is explored from the accessible models. These pre-trained models are usually trained on massive datasets. A model trained on a similar dataset is usually selected from a pool of available models. Secondly, the selected model with saved weights is reapplied. It is used as the beginning point to prepare the desired model for the second assignment of interest. This may include utilizing all or some parts of the model. Thirdly, the obtained model is fine-tuned and modified as per the undertaking of interest.

The *ImageNet* project is an extensive repository of various images and models designed for use in visual object recognition. The various models trained on the ImageNet project are available for research and re-application. When these learned models are used to solve similar problems, this is termed transfer learning. In the current paper, we have used ResNet50, MobileNet, and Inception Net for binary classification.

ResNet (https://neurohive.io/en/popular-networks/resnet/), i.e., Residual Networks, is a neural network utilized for numerous computer vision-based exercises. This model was the winner of the ImageNet challenge in 2015—incredible advancement with ResNet permits to effectively prepare top to bottom organizations with 150 + layers. Earlier, ResNet usage for the profound neural network was troublesome because of the vanishing of inclinations. ResNet then presented the idea of skip association. ResNet skip associations are utilized in model structures like fully convolutional network (FCN) and U-Net. They are utilized

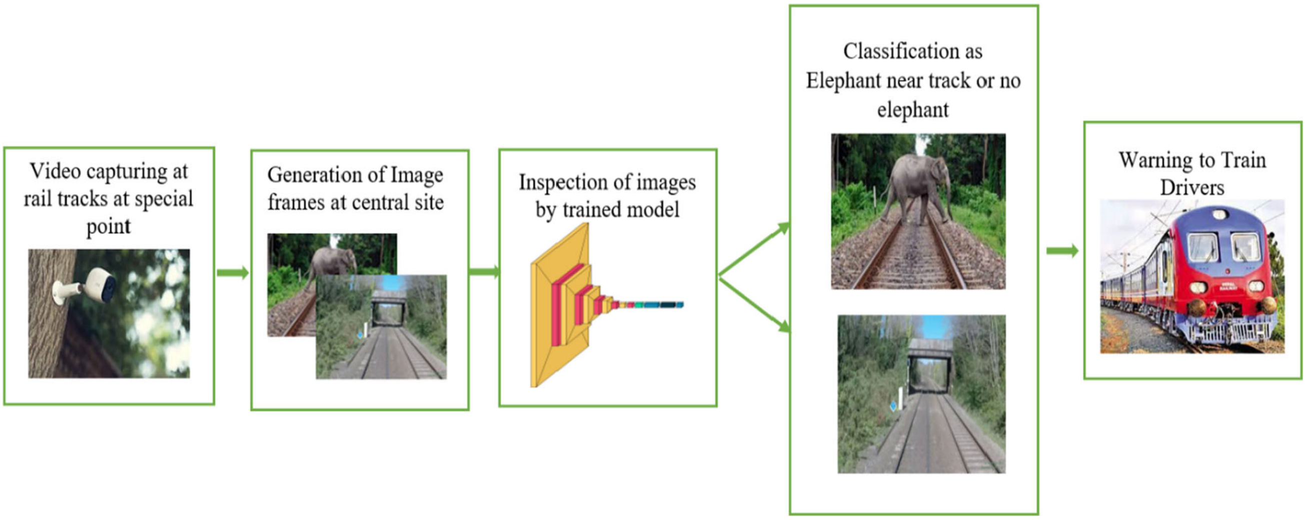

Fig. 3 Proposed model for the prevention of train–elephant collision

to stream data from prior layers to later layers in the model. In these designs, data are passed from the downsampling layers to the upsampling layers. The ResNet-50 model consists of 5 stages, each with a convolution and identity block. The convolution and identity block has three convolution layers each. The ResNet-50 has over 23 million trainable parameters.

MobileNets (Howard et al. 2017) are a class of small, low-computation models that can be utilized for order, recognition, and other regular tasks which can be undertaken by convolutional neural networks. These are viewed as incredible deep learning models to be utilized on cell phones considering their small size. Presently, while MobileNets are quicker and more modest than other significant organizations, as VGG16, for instance, there is a trade-off. That trade-off is precision. Although, MobileNets are not as precise as other huge counterparts; however, they still really perform exceptionally well.

Inception/GoogleNet (Szegedy et al. 2016) Google devised a module called the inception module that approximates a sparse CNN with standard dense construction. Since only a small number of neurons are adequate, the width/number of the convolutional filters of a particular kernel size is kept small. Additionally, it uses convolutions of different sizes to capture details at varied scales (5×5, 3×3, 1×1). It exploits the fact that most activations in a deep network are either needless (value of zero) or unnecessary because of correlations between them. Consequently, the most efficient architecture of a deep network will have a sparse connection between the activations, which implies that all 512 output channels will not connect with all the 512 input channels. Another salient point about the module is that it has a bottleneck layer of 1×1 convolutions. It helps in the massive reduction of the computation requirement.

5 Proposed model for the prevention of train elephant collision

We have proposed a complete surveillance model for the prevention of train elephant collisions in Fig. 3. Initially, video cameras will be implanted at vulnerable sites. The video from these sites will be converted to frames and are sent to a central site. A trained deep vision model will be kept at central sites. This model will inspect the extracted frames for the presence of elephants near rail tracks. If such a situation is identified, the alarm will be triggered in trains near that track, and a warning sound will be generated at the track to warn the Elephant.

For the visual inspection, several deep learning architectures have been experimented with. First, a new CNN architecture from scratch is specially designed for this application. Many combinations for various layers, optimizers, and activation functions are tried to get the best results. Experimentation with transfer learning is conducted using three existing deep learning architectures from ImageNet, i.e., ResNet50, MobileNet, and Inception V3. All the experiments are done using Keras and Tensorflow in Google Collaboratory. The details of the experiment and its implementation are discussed in the following sections.

5.1 Details of proposed CNN architecture

The proposed architecture has five convolutional layers and five pooling layers. Then the output is flattened, and a fully

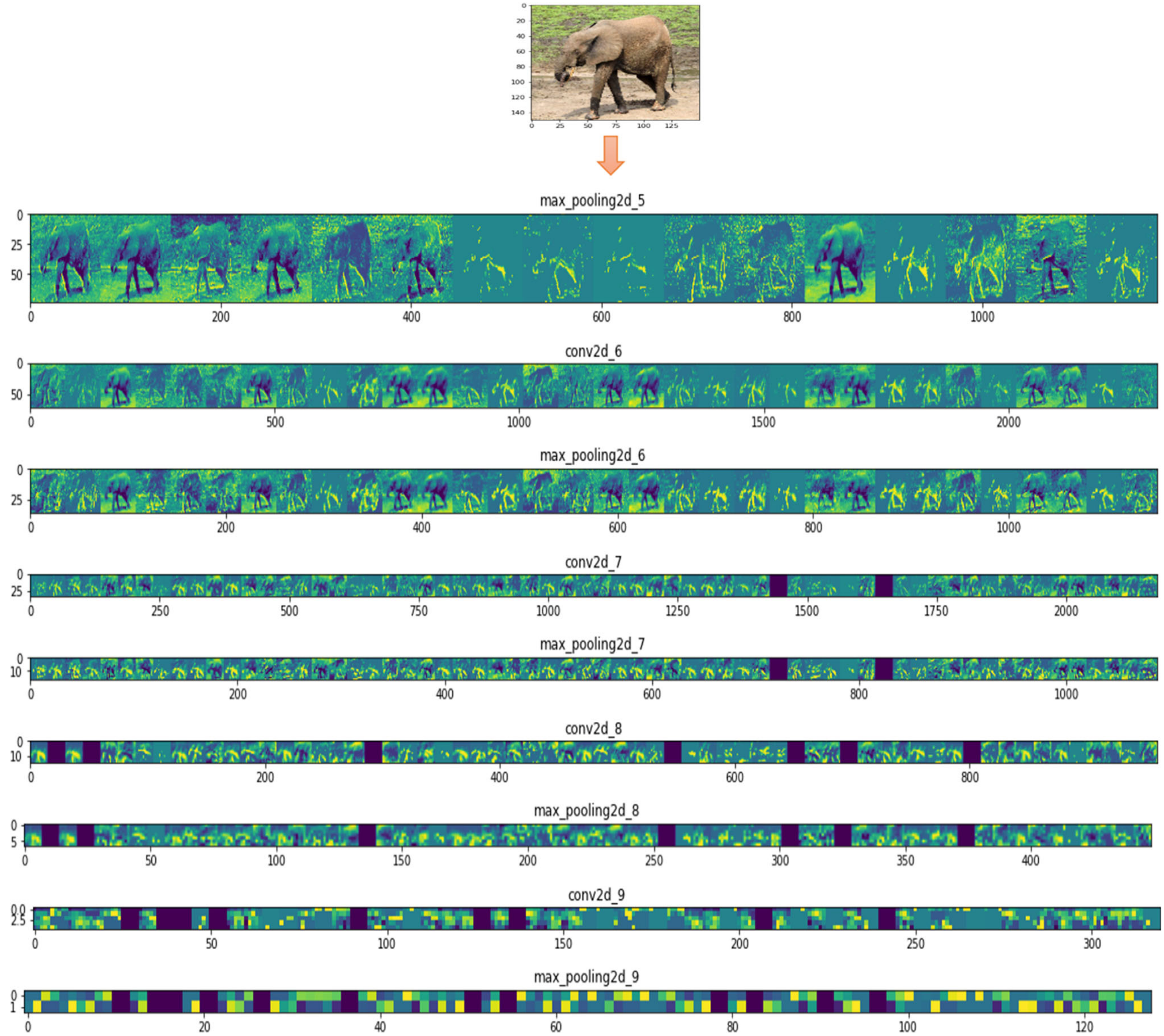

Fig. 4 Analysis of output of convolution layer of the proposed CNN architecture

connected layer is employed for binary classification. Figure 4 shows the output of convolution layers and pooling layers as a heat map. Figure 5 shows the details of the proposed CNN model for classifying images of rail track with or without an elephant. Various components used in the proposed CNN are shown in detail. Data augmentation is used so that model could become translation, rotation, and scale-invariant. A batch size of 32 is used for training as well as testing. Due to limitations in computing resources, the model is trained for 50 epochs. RMSprop optimizer is used, which is like the gradient descent algorithm with momentum. The RMSprop optimizer restricts the oscillations in the vertical direction. Thus, the algorithm could take more significant steps in the horizontal direction and can converge faster. Binary_crossentropy is used as a loss function. It is a measure of the accuracy of the model in predicting the desired object in the image. The "Sigmoid" function is used for binary classification. Hence, it is used to categorize the image with and without an elephant. Dropout is used to limit the parameters in fully connected layers. Figure 6a illustrates the architecture of the proposed CNN.

5.2 Details of proposed inception network architecture

Several deep architectures like ResNet, MobileNet, VGG, and InceptionNet models are provided by the ImageNet community to be used for other applications. We have done experiments with Resnet, MobileNet, and InceptionNet

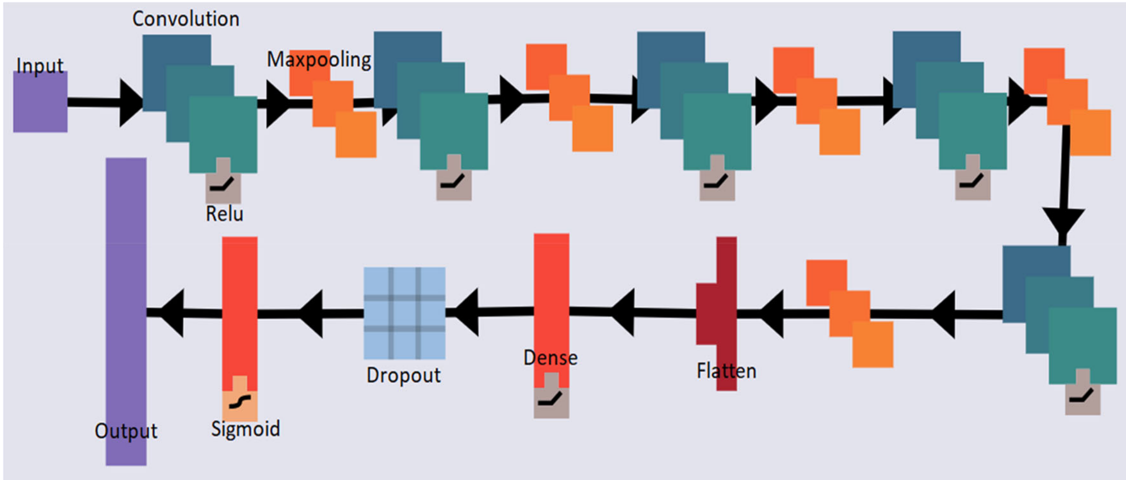

Fig. 5 The detailed architecture of the proposed CNN

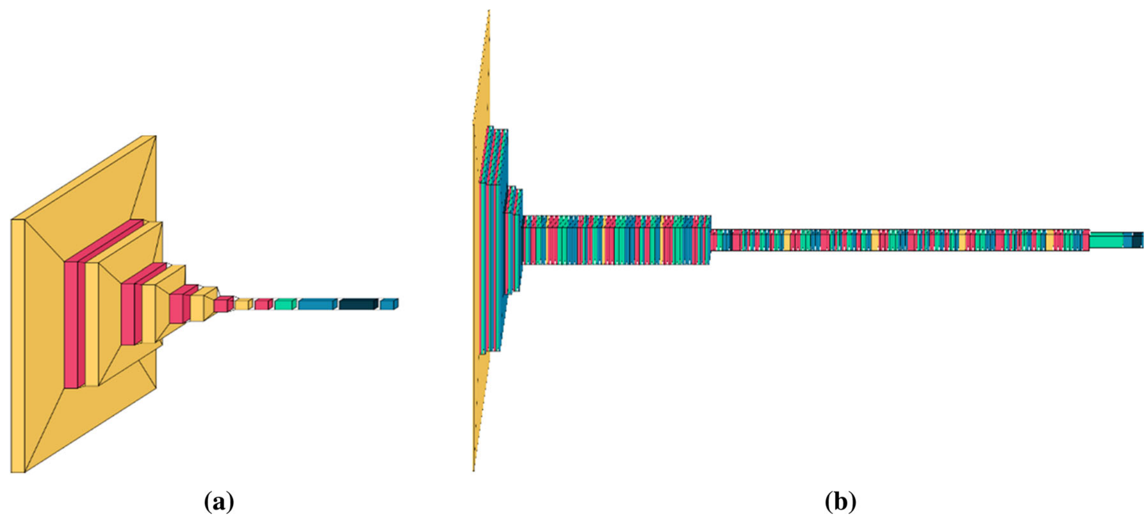

Fig. 6 Proposed **a** CNN **b** InceptionNet architecture

Table 3 Comparison of different models experimented in current work

Model	Layers	Parameters	Accuracy	Precision	Recall	F1-score
Proposed CNN	7	2 lakh	99.53%	1.0	0.99	1.0
Resnet50	50	23 million	99.81%	1.0	1.0	1.0
MobileNet	28	4 million	99.81%	1.0	1.0	1.0
Inception V3	22	5 million	99.91%	1.0	1.0	1.0

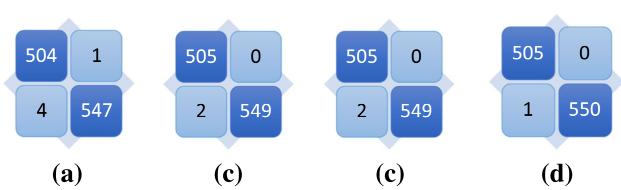

Fig. 7 Confusion matrix **a** proposed CNN, **b** ResNet50 **c** MobileNet **d** Inception V3

models and found that InceptionNet has done remarkably well in identifying the image with and without elephants. The Inception architecture has 22 convolutional layers for the extraction of the feature. The architecture of the Inception network is illustrated in Fig. 6b. The last layers are customized to give binary classification for elephant detection. Parameters, similar to the proposed CNN model, are used for the Inception model. A batch size of 32 is used for training as well as testing. Model is trained for 50

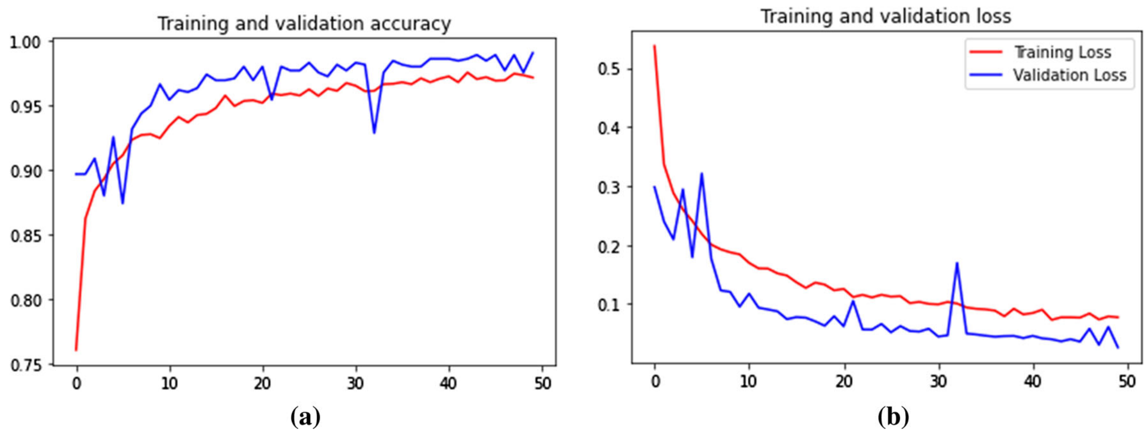

Fig. 8 Training and validation **a** accuracy and **b** Loss for proposed CNN model

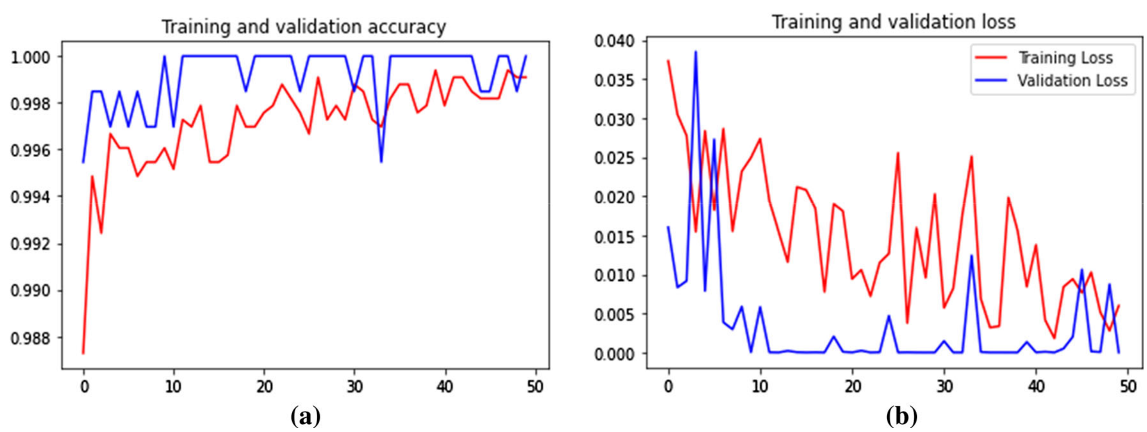

Fig. 9 Training and validation **a** accuracy and **b** Loss for the Inception V3 model

Table 4 Comparison with the existing state of art

Method	Dataset size	FPR	Accuracy
Zeppelzauer et al. (2005)	715	2.5%	91.7%
Ravikumar et al. (2018)	1792	3.1%	92.7%
Proposed CNN	4305	0.18%	99.53%
Proposed Inception v3	4305	0	99.91%

epochs. RMSprop optimizer is used, which is like the gradient descent algorithm with momentum. Binary_crossentropy is used as a loss function. The "Sigmoid" function is used for binary classification to categorize the image with and without an elephant. Dropout is used to limit the parameters in fully connected layers.

6 Results and discussion

As mentioned in the methodology, different CNN and transfer learning models have been experimented with for binary classification. Table 3 shows the results and comparison between the proposed models. The proposed CNN model achieved an accuracy of 99.53% and is computationally effective with only seven layers and about two lakh parameters. Transfer learning-based models have performed better as they have more layers and more parameters and hence extensive learning. Resnet50 and MobileNet performed better with an accuracy of 99.81%. Inception Net has performed best in binary classification and achieved an accuracy of 99.91%. The inception Model has a relatively lesser number of layers and parameters and is suited best for the current problem.

Figure 7 shows the confusion matrix for the test dataset for 1056 images. Five hundred five images contain elephant (s) in frames, and 551 images have rail tracks without an elephant. Figure 7a shows the confusion matrix for the proposed lightweight CNN. Figure 7b–d shows the confusion matrix for transfer learning models. For the

Fig. 10 Result of Inception V3 model on unknown 40 images from the Internet

proposed CNN, only five images out of 1056 are miscategorized. The performance of TL models is good. Only two images are misclassified for Resnet50 and MobileNet models. The performance of the Inception model is best with only one image misclassified as a frame with an elephant. The best part is that the model never failed in detecting an elephant when it is there, which is imperative.

Figure 8a, b shows the training and validation accuracy and loss for the proposed CNN, and Fig. 9a, b shows the same for the best transfer learning model, i.e., the Inception model. The trend in the graph shows that validation accuracy and loss have followed the training accuracy and loss trend.

Table 4 shows the comparison of the proposed model with similar work. Zeppelzauer et al. (Farah et al. 2011) proposed a multi-modal early warning system to detect elephants in wild video recordings. Elephants are identified based on the color model of their body. The SVM classifier was used to predict the presence of an elephant in the wild video recordings. The dataset used had 715 images, and the accuracy obtained is 91.7%. Ravikumar et al. (2019) proposed MobileNet architecture for single-shot detection of elephants in wild videos. Their CNN model achieved an accuracy of 92.7%. Our proposed CNN and tuned Inception model outperformed the existing models with better accuracy of 99.53% and 99.91%, respectively, with negligible false positive ratio. Therefore, the proposed model is convincingly better than the existing model and is ready for use in real-time scenario.

6.1 Testing beyond dataset

When the testing is done for images with and without elephant on track, accuracy obtained is almost 100%. However, we experimented with complex images with humans and small animals on track to check the performance of the proposed model in a real-time situation. These images are not part of the training. Figure 10 shows the results obtained by the Inception V3 model for typical 40 real-time images found on the Internet. It is observed that the small animals are not misunderstood as elephants, as shown in Row 2–3, and hence the model is ready to be used in an actual situation. For few images in Row 1 containing big animals are misclassified as an image with an elephant. The poor resolution of these images is also one of the significant constraints. This can undoubtedly be improved with a larger, ideal dataset which is not available as of now. The best part is that the elephant present nearby is always identified with 100% accuracy.

7 Conclusion

As human activities had led to intervention in wildlife, its consequences are visible in terms of animal extinction. Nowadays, many efforts are being made for animal conservation, and advanced technology can certainly help in this regard. Computer vision and the Internet of things together can do a lot in this direction. Along the same lines, this paper aims to develop a model for HEC prevention. It detects the elephants on/near rail track using a deep vision model. Several models based on CNN and TL are experimented with and compared for effort and efficiency. The Inception v3 model has performed best for this application because of its high accuracy and zero negative rates. The model can detect the presence of elephants and hence generate alarms on-site and send warning signals to nearby trains. In the future, a dedicated dataset and model can be prepared and trained for the detection of a wide range of animals so that accurate classification and identification can be done to preserve wildlife. In addition, the proposed architectures can be experiment for different applications oriented to object detection (Gupta et al. 2019).

Authors' Contributions All authors contributed to the review of literature, the study of methodologies, design, and implementation of the proposed method. Implementation, Material preparation, literature review, and analysis were performed by Surbhi Gupta, Neeraj Mohan, Madhavi Karanam, and Krishna Chythanya. Dr. Padmalaya Nayak has contributed to the second draft of the manuscript. She has added literature related to the problem undertaken. She helped in redrawing Fig. 3 and added Fig. 5 for the overall improvement of the manuscript. All authors commented on previous versions of the manuscript and helped in draft of second version. All authors have thoroughly read, commented, and approved the final manuscript.

Declarations

Conflict of interest There is no conflict of interest by any of the authors.

Supplementary Information

The online version contains supplementary material available at https://doi.org/10.1007/s00500-021-06493-8.

References

Ardovini A, Cinque L, Sangineto E (2008) Identifying elephant photos by multi-curve matching. Pattern Recogn 41(6):1867–1877

Backs JAJ, Nychka JA, Clair CS (2017) Warning systems triggered by trains could reduce collisions with wildlife. Ecol Eng 106:563–569

Beery S, Van Horn G, Perona P (2018) Recognition in terra incognita. In: Proceedings of the European conference on computer vision (ECCV), pp 456–473

Bíl M, Andrášik R, Dul'a M, Sedoník J (2019) On reliable identification of factors influencing wildlife-vehicle collisions along roads. J Environ Manage 237:297–304

Burghardt T, Ćalić J (2006) Analysing animal behaviour in wildlife videos using face detection and tracking. IEE Proc vis Image Signal Process 153(3):305–312

Chen G, Han TX, He Z, Kays R, Forrester T (2014) Deep convolutional neural network-based species recognition for wild animal monitoring. In: 2014 IEEE international conference on image processing (ICIP). pp 858–862. IEEE

Chythanya K, Madhavi K, Ramesh G (2020) A Machine learning enabled IoT device to combat elephant mortality on railway tracks. In: Springer Proceedings of 2nd international conference on innovative data communication technologies and applications (ICIDCA 2020)-Sept. 2020. In press

Devost E, Lai S, Casajus N, Berteaux D (2019) Fox Mask: a new automated tool for animal detection in camera trap images. BioRxiv, p 640037

Dhanaraj JSA, Kumar Sangaiah A (2018) Elephant detection using boundary sense deep learning (BSDL) architecture. J Exp Theor Artif Intell, pp 1–16

Farah R, Langlois JP, Bilodeau GA (2011) Rat: robust animal tracking. In: 2011 IEEE international symposium on robotic and sensors environments (ROSE), pp 65–70. IEEE

Gadekallu TR, Rajput DS, Reddy MPK, Lakshmanna K, Bhattacharya S, Singh S, Jolfaei A, Alazab M (2020) A novel PCA–whale optimization-based deep neural network model for classification of tomato plant diseases using GPU. J Real-Time Image Process, pp 1–14

Gadekallu TR, Alazab M, Kaluri R, Maddikunta PKR, Bhattacharya S, Lakshmanna K, Parimala M (2021) Hand gesture classification using a novel CNN-crow search algorithm. Complex Intell Syst, pp 1–14

Gupta S, Kumar M, Garg A (2019) Improved object recognition results using SIFT and ORB feature detector. Multimed Tools Appl 78(23):34157–34171

Hannuna SL, Campbell NW, Gibson DP (2005) Segmenting quadruped gait patterns from wildlife video

Howard AG, Zhu M, Chen B, Kalenichenko D, Wang W, Weyand T, Andreetto M, Adam H (2017) Mobilenets: efficient convolutional neural networks for mobile vision applications. arXiv preprint arXiv:1704.04861

Iwendi C, Srivastava G, Khan S, Maddikunta PKR (2020) Cyberbullying detection solutions based on deep learning architectures. Multimed Syst, pp 1–14

Jayakumar R, Swaminathan R, Harikumar S, Banupriya N, Saranya S (2020) Animal detection using deep learning algorithm. J Crit Rev 7(1):434–439

Kellenberger B, Marcos D, Tuia D (2018a) Detecting mammals in UAV images: best practices to address a substantially imbalanced dataset with deep learning. Remote Sens Environ 216:139–153

Kellenberger B, Marcos D, Lobry S, Tuia D (2019) Half a percent of labels is enough: efficient animal detection in UAV imagery using deep CNNs and active learning. IEEE Trans Geosci Remote Sens 57(12):9524–9533

Kellenberger B, Volpi M, Tuia D (2017) Fast animal detection in UAV images using convolutional neural networks. In: 2017 IEEE international geoscience and remote sensing symposium (IGARSS) (pp 866–869). IEEE

Kellenberger B, Marcos D, Tuia D (2018) Best practices to train deep models on imbalanced datasets—a case study on animal detection in aerial imagery. In: Joint European conference on machine learning and knowledge discovery in databases. pp 630–634. Springer, Cham

Koik BT, Ibrahim H (2012) A literature survey on animal detection methods in digital images. Int J Future Comput Commun 1(1):24

Korschens M, Denzler J (2019) Elpephants: a fine-grained dataset for elephant re-identification. In: Proceedings of the IEEE international conference on computer vision workshops, pp 0–0

Kumar M, Gupta S, Gao XZ, Singh A (2019a) Plant species recognition using morphological features and adaptive boosting methodology. IEEE Access 7:163912–163918

Kumar S, Baline HV, Sivakumar T, Potluri VP (2019b) detection of wild elephants using image processing on raspberry PI3. Int J Comput Sci Mobile Comput 8(2):104–115

Langbein J (2011) Monitoring reported deer road casualties and related accidents in England to 2010. Research Report 2011/3. The Deer Initiative, Wrexham, UK

Mammeri A, Zhou D, Boukerche A, Almulla M (2014) An efficient animal detection system for smart cars using cascaded classifiers. In: 2014 IEEE international conference on communications (ICC), pp 1854–1859. IEEE

Mandal RK, Bhutia DD (2018) A proposed artificial neural network (ANN) model using geophone sensors to detect elephants near the railway tracks. Advanced computational and communication paradigms. Springer, Singapore, pp 1–6

Marais JC (2018) Automated elephant detection and classification from aerial infrared and colour images using deep learning (Doctoral dissertation, Stellenbosch: Stellenbosch University)

Mönck HJ, Jörg A, von Falkenhausen T, Tanke J, Wild B, Dormagen D, Piotrowski J, Winklmayr C, Bierbach D, Landgraf T (2018) BioTracker: an open-source computer vision framework for visual animal tracking. arXiv preprint arXiv:1803.07985

Morse G, Liu T, Gilchrist A, Halliday N, Heavisides J, Hopper D, McKay S, Nowell R, Pitman S, Woods M (2014) Analysis of the risk from animals on the line – Issue 2. Report. Rail Safety and Standards Board, London, UK

Naude J, Joubert D (2019). The aerial elephant dataset: a new public benchmark for aerial object detection. In: Proceedings of the IEEE conference on computer vision and pattern recognition workshops, pp 48–55

Norouzzadeh MS, Nguyen A, Kosmala M, Swanson A, Palmer MS, Packer C, Clune J (2018) Automatically identifying, counting, and describing wild animals in camera-trap images with deep learning. Proc Natl Acad Sci 115(25):E5716–E5725

Patman J, Michael SC, Lutnesky MM, Palaniappan K (2018) Biosense: real-time object tracking for animal movement and behavior research. In 2018 IEEE Applied Imagery Pattern Recognition Workshop (AIPR), pp 1–8. IEEE

Praczyk T (2020) Neural collision avoidance system for biomimetic autonomous underwater vehicle. Soft Comput 24(2):1315–1333

Raja MAA, Ramya MK, Kousalya MB, Jeeva MS (2018) Prevention of wild animals from accidents using image detection and edge algorithm. PREVENTION, 5(11)

Ramanan D, Forsyth DA (2003) Using temporal coherence to build models of animals. p 338. IEEE

Ravikumar S, Vinod D, Ramesh G, Pulari SR, Mathi S (2020) A layered approach to detect elephants in live surveillance video streams using convolution neural networks. J Intell Fuzzy Syst 38(5):6291–6298

Redmon J, Farhadi A (2017) YOLO9000: better, faster, stronger. In: Proceedings of the IEEE conference on computer vision and pattern recognition. pp 7263–7271

Ren S, He K, Girshick R, Sun J (2015) Faster r-cnn: towards real-time object detection with region proposal networks. Adv Neural Inf Process Syst 28:91–99

Rey N, Volpi M, Joost S, Tuia D (2017) Detecting animals in African Savanna with UAVs and the crowds. Remote Sens Environ 200:341–351

Schneider S, Taylor GW, Linquist S, Kremer SC (2019) Past, present and future approaches using computer vision for animal re-identification from camera trap data. Methods Ecol Evol 10(4):461–470

Sharkawy AN, Koustoumpardis PN, Aspragathos N (2020) Human–robot collisions detection for safe human–robot interaction using one multi-input–output neural network. Soft Comput 24(9):6687–6719

Sharma SU, Shah DJ (2016) A practical animal detection and collision avoidance system using computer vision technique. IEEE Access 5:347–358

Shukla P, Dua I, Raman B, Mittal A (2017) A computer vision framework for detecting and preventing human-elephant collisions. In: Proceedings of the IEEE international conference on computer vision workshops, pp 2883–2890

Sugumar SJ, Jayaparvathy R (2014) An improved real time image detection system for elephant intrusion along the forest border areas. Sci World J

Szegedy C, Vanhoucke V, Ioffe S, Shlens J, Wojna Z (2016) Rethinking the inception architecture for computer vision. In: Proceedings of the IEEE conference on computer vision and pattern recognition, pp 2818–2826

Tanwar M, Shekhawat NS, Panwar S (2017) A survey on algorithms on animal detection. Int J Future Revolut Comput Sci Commun Engineering. 3(6):33–35

Tweed D, Calway A (2002) Tracking multiple animals in wildlife footage. In: Object recognition supported by user interaction for service robots, vol 2, pp 24–27, IEEE

Venkataraman AB, Saandeep R, Baskaran N, Roy M, Madhivanan A, Sukumar R (2005) Using satellite telemetry to mitigate Elephant–human conflict: an experiment in northern West Bengal, India. Current Science, pp 1827–1831

Vermeulen C, Lejeune P, Lisein J, Sawadogo P, Bouché P (2013) Unmanned aerial survey of elephants. PLoS ONE 8(2):e54700

Zendel O, Murschitz M, Zeilinger M, Steininger D, Abbasi S, Beleznai C (2019) Railsem19: a dataset for semantic rail scene understanding. In: Proceedings of the IEEE conference on computer vision and pattern recognition workshops, pp. 0–0

Zeppelzauer M (2013) Automated detection of elephants in wildlife video. EURASIP J Image Video Process 2013(1):46

Zeppelzauer M, Stoeger AS (2015) Establishing the fundamentals for an elephant early warning and monitoring system. BMC Res Notes 8(1):409

Zotin AG, Proskurin AV (2019) Animal detection using a series of images under complex shooting conditions. Int Arch Photogramm Remote Sens Spat Inf Sci