

Deep Vision Based Surveillance System to Prevent Train-Elephant Collisions

Surbhi Gupta (✉ royal_surbhi@yahoo.com)

Gokaraju Rangaraju Institute of Engineering and Technology <https://orcid.org/0000-0003-0618-8369>

Neeraj Mohan

Punjab Technical University: IK Gujral Punjab Technical University Jalandhar

Krishna Chythanya Nagaraju

Gokaraju Rangaraju Institute of Engineering and Technology

Madhavi Karanam

Gokaraju Rangaraju Institute of Engineering and Technology

Research Article

Keywords: Human-Elephant Collision, Rail Track Monitoring, Deep Vision, Data Augmentation, Transfer Learning

Posted Date: April 26th, 2021

DOI: <https://doi.org/10.21203/rs.3.rs-192950/v1>

License: © ⓘ This work is licensed under a Creative Commons Attribution 4.0 International License.

[Read Full License](#)

Version of Record: A version of this preprint was published at Soft Computing on November 12th, 2021. See the published version at <https://doi.org/10.1007/s00500-021-06493-8>.

DEEP VISION BASED SURVEILLANCE SYSTEM TO PREVENT TRAIN-ELEPHANT COLLISIONS

Surbhi Gupta¹, Neeraj Mohan², Krishna Chythanya Nagaraju³, Madhavi Karanam⁴

^{1,3,4}Deptt. of CSE, Gokaraju Rangaraju Institute of Engineering and Technology, Hyderabad

²Deptt. of CSE, IK Gujral Punjab Technical University, Mohali

royal_surbhi@yahoo.com¹

erneerajmohan@gmail.com²

kcn_be@rediffmail.com³

madhaviranjana@griet.ac.in⁴

Corresponding author: Dr. Surbhi Gupta

Abstract:

Animal conservation is imperative, and a lot of technology has been used in different ways. The endangered species like tiger and elephant has raised the need for such efforts. Human-Elephant Collision (HEC) has been an active area of research but still, the optimum solution is not found. As trains are widely used transportation medium in Asian countries, the rail track is even laid down through forest areas and hence intervene the wildlife. Elephants due to their bulky size often become victims of trains. Such tragedy is common especially in green belts in southern zones of India. To rectify the problem, we have proposed a deep vision-based model to identify the elephant near-site using implanted video cameras. Four different models are proposed for the identification of elephants in image/video. One novel lightweight CNN based model is proposed. Three Transfer Learning (TL) models, i.e., ResNet50, MobileNet, Inception V3 have been experimented and tuned for elephant detection. These highly accurate and precise models can alarm the trains hence it can save a precious life.

Keywords: Human-Elephant Collision, Rail Track Monitoring, Deep Vision, Data Augmentation, Transfer Learning

I. INTRODUCTION

Animal mortality is becoming a major concern day by day in the whole world as it is disturbing ecological balance and in certain cases, even the species are being endangered. The International Union for Conservation of Nature (IUCN) has already given endangered status to the Indian Elephant. Particularly in India, human life is intrinsically entangled with the big animal Elephant. Whether it is culture, mythology, or the Hindu custom, Elephant plays a very major role and is also considered as a sacred animal besides the symbol of intellectual strength. According to Hinduism, Lord Ganesh is having an Elephant head and is considered as the obstacle remover in life. Hence, every auspicious work usually starts with a prayer to the Elephant faced Ganesh. Whereas, in contrast, the life of elephants is full of struggle in the present scenario. Due to a lot of human intervention into the habitats of these, it is an endangered species now. Whether it is the costly ivory tusk of the mammal or the deforestation by the greedy human beings, at the receiving end are the gigantic mammal's Elephants [1,2].

Humans have interfered with their habitat of wilds. Due to the requirement of space for living, agriculture, and transport, a lot of forests have been cleared. Due to the transport needs a lot of roads and rail tracks exist in/near the green zone or forest areas. In India, trains are the cheapest and mass transport mechanism to connect each corner of land for better business prospects. And usually, these tracks pass through a lot of forest occupied regions and hence a danger for wild animals. 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 end up losing their life. Many steps have been taken to avoid the clash. Various alert systems and barriers have been placed at crucial sites but still, the system is taking a huge toll on the life of elephants. The Human-Elephant collision has become a big research point owing to a declining number of Elephants. Many researchers have worked on the impact of railway tracks being passed through the forest area. A lot of researchers studied the statistics of elephants killed by train collisions over the past few years [3] as well as measures taken by the Govt. of India in handling the issue meticulously. This paper can be considered as an extension of the proposal made in [3]. The advent in image processing can be effectively utilized in avoiding a human-elephant collision or train elephant collision. Image processing powered with machine learning and deep neural networks has given promising results in all real-time applications. Recently, multilayer feedforward neural network-based approach is proposed for human-robot collision detection [4]. Similarly, collision avoidance system is proposed for biomimetic autonomous underwater vehicle using artificial neural network [5]. Hence, the use of deep learning for the detection of human-elephant collision can be explored for avoidance for the same. Thus, the main contributions of this paper are:

1. To review the existing efforts for Human Elephant Collision prevention
2. To propose a lightweight 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, MOBILE NET, and INCEPTION V3 for precise identification of elephant on track
5. A complete model for alarm generation to avoid HEC and hence animal conservation.

II. RELATED LITERATURE

A good number of researchers have worked on image identification of elephants that help in controlling

elephant train collision system development. Many authors worked on animal detection algorithms using machine learning and deep learning. Koik and Ibrahim [6] presented a literature survey on animal detection methods in digital images. Tanwar et al., [7] presented a survey on algorithms on animal detection. A small summary of contributions made by different researchers in this area is shown in Table 1 below.

Table1. Summary of contribution for prevention of Human-Animal/Elephant Collision

Approaches applied for	Author	Proposed Method
Animal detection and tracking	Tweed and Calway, 2002[8]	Movement track model using condensation particle filtering.
	Ramanan and Forsyth, 2003[9]	Temporal Coherency based model for animal detection
	Hannuna et al., 2005[10]	Animal gait identification model
	Burghardt and Čalić, 2006 [11]	Haar-like feature extraction with Adaboost Classifier based model
	Farah et al., 2011[12]	A robust method to track animals and determine their motion pattern
	Mammeri et al., 2014[13]	Two-step classification system using LBP-Adaboost followed by HOG-SVM Classifier
	Sharma and Shah, 2016[14]	Animal detection and collision avoidance system using Computer Vision
	Norouzzadeh and Nguyen, 2018 [15]	Automatically identifying, counting, and describing wild animals in camera-trap images with Deep Learning
	Raja et al., 2018[16]	Image identification using an edge detection algorithm
	Devost et al., 2019[17]	Automated tool for animal detection in camera trap images
	Zotin and Proskurin, 2019 [18]	Animal detection using a series of images under complex shooting conditions.
	Backs et al., 2017 [19]	Hebb's Law of a learning-based model
	Bill et al., 2019 [20]	Kernel Density Estimation based model
	Jayakumar et al., 2020[21]	Animal detection using a deep learning algorithm
Elephant detection and tracking	Venkataraman et al., 2005[22]	Satellite image-based elephant position of African elephant
	Ardovini et al., 2008[23]	Ear shape-based model for an elephant identification system
	Vermeulen et al., 2013[24]	Unmanned Aerial Vehicles at a height of 100mts for tracking
	Zeppelzauer, 2013[25]	Automated detection of elephants in wildlife video
	Sugumar and Jayaparvathy, 2014[26]	Image feature extraction and similarity matching model based on Euclidian and Manhattan distance
	Zeppelzauer et al., 2015 [27]	Color model-based elephant detection
	Shukla et al., 2017 [28]	Computer vision framework for detecting and preventing human-elephant collisions
	Mandal et al., 2018 [29]	Unsupervised Artificial Neural Network model using Geophone Sensors to detect elephants
	Marais et al., 2018 [30]	Automated elephant detection and classification from aerial infrared and colour images using deep learning
	Dhanaraj and Sangaiah et al., 2018 [31]	Elephant detection using boundary sense deep learning (BSDL) architecture
	Kumar et al., 2019[32]	Detection of Wild elephants using image processing on Raspberry Pi3
	Ravikumar et al., 2020[33]	Transfer Learning based MobilNet model for elephant detection

Several approaches for animal detection and tracking have been proposed [8-21]. Sharma and Shah [14] proposed real-time animal detection and collision avoidance system using a computer vision technique. Norouzzadeh and Nguyen [15] performed the identification and counting for wild animals in camera-trap images with deep learning. Raja et al., [16] studied the prevention of wild animals from accidents using image detection and edge algorithm. Devost et al., [17] proposed a new automated tool for animal detection in camera trap images.

Zotin and Proskurin [18] performed animal detection using a series of images under complex shooting conditions. Backs et al., 2017 [19] proposed low-cost electronic-based devices to avoid train collision with animals on track. They performed testing on Bayy National Park and Yoho National Park where grizzly bears, Black bears, Wolves, and Moose were found to be killed in large numbers. The first method makes use of two paired but placed distantly devices and of which, one device takes care of detecting passing train and sends information to the paired warning device placed at a high striking rate position. The second method includes predicting train arrival time at a distance hypothetically considered as 200 meters and activates an integrated warning mechanism at the desired time. Random Forest classification models were used with 10-fold 10-repeat cross-validation and 80% detection rate was claimed. Bill et al., 2019 [20] claimed that when the wildlife-vehicle collisions (WVC) are categorized into their occurrences in the cluster and outside the cluster, achieving statistically significant local factors were observed. The Kernel Density Estimation (KDE) method extended with Monte Carlo simulation called KDE+ was used to identify more significant clusters. They achieved a 95% confidence interval and Road width; the presence of shrubs and habitat type are observed as the most significant variables. Jayakumar et al., [21] proposed animal detection using a deep learning algorithm.

Other authors have made efforts for elephant detection and tracking specifically [22-33]. Initially, feature extraction was used to detect elephants in frames. Sugumar and Jayaparvathy, 2014 [26] used images captured with a camera mounted on towers or trees and being 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 find the difference with a threshold of 5 images to be matched for the query image to send an SMS. The researchers made use of an optimized distance measure that gave an 18.5% better performance as compared to other measures. Zeppelzauer [27] implemented the detection of elephants in wildlife videos. Shukla et al., 2017 [28] 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 made use of Deep Conventional Neural Networks to recognize Elephants as well as Elephant Faces and track the moment of animal in the case out of conflict with a warning message in case of conflict occurring. The researchers made use of the transfer learning model Alex-Net by fc7 layer with 4096*1 dimension. This representation is fed to support vector machines for binary classification. The work was carried out using a 12 videos dataset comprising of 500 images of elephants. The researchers claim to achieve a 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 [29] 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 [30] proposed automated elephant detection and classification using aerial infrared and color images using deep learning. Similarly, Dhanaraj and Sangaiah [31] did elephant detection use boundary sense deep learning (BSDL) architecture. Kumar et al. [32] performed the detection of Wild elephants using IOT based solution. Ravikumar et al., [33] proposed MobileNet architecture for successful elephant detection. Although a lot of methods [8-33] were proposed for animal and elephant detection, the usage of CNN architecture and Transfer Learning (TL) for the same is still not explored fully. This paper aims to employ CNN and transfer learning for efficient detection of elephants in different positions on rail tracks so that alert systems can be implemented.

III. DATASET USED

As the experiment aims at identifying elephants on or near the rail track, we have searched for various images and datasets which serve the purpose. Naude and Joubert [34] 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 very difficult to capture natural images of an elephant on or near the track. Some random images are available though. Due to this limitation, we have decided to use a dataset of elephants and a dataset with rail track images. Two public datasets are used in this current research i.e. ELPephant [35] and RailSem19 [36].

ELPephant dataset is a huge dataset of 2078 images with the image of single and multiple elephants. Moreover, different classes of elephants are covered with 3 to 22 images for each. Different features and different poses of elephants are included. Left, right, and front profile are included. Sitting elephants are also featured in some images. This dataset was specifically designed for the class detection of the elephants. Moreover, the images are collected over 15 years, and hence the dataset is unique and very useful in the identification of elephants in an image/video. As the experiment aims at identifying the

elephants at the track, we need images with clear rails tracks too. Therefore, we have used the dataset RailSem19 as it contains a lot of 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 above-mentioned dataset, we needed more practical images to test the proposed model. Therefore, we used different keywords like “elephant on rail track”, “animals on rails track”, etc., to search for some real 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 high variance in images during the training phase. Fig. 1 shows the sample images taken for experimentation. Row 1 shows the sample images from the ELPephaunt dataset. Row 2 shows the images from the RailSem19 dataset. Row 3 and 4 show random images based on google search with different keywords “animals on rail track” and “elephant on rail track”, respectively.

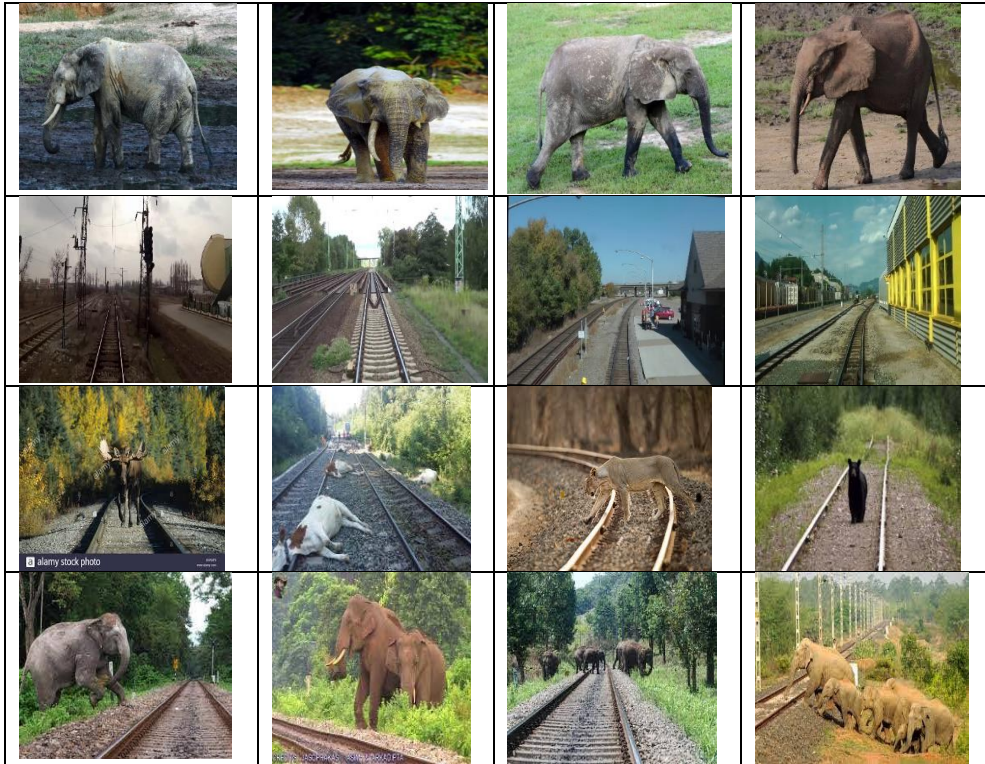


Fig. 1: Sample images used for training and testing

Row 1: images from Elephant dataset; Row 2: Images from RailSem19 dataset; Row 3: Images from internet with the keyword “Animals on Rail track”; Row 4: Images from internet with the keyword “Elephants on Rail track”

Table 2: Total number of images considered for experimentation

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

IV. PROPOSED MODEL FOR THE PREVENTION OF TRAIN ELEPHANT COLLISION

We have proposed a complete surveillance model for the prevention of train elephant collision. Initially, video cameras will be implanted at vulnerable sites.

The video from these sites will be converted to frames at 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.

Fig. 2 illustrates the complete model for train elephant collision prevention. Visual Inspection by Trained models in Fig. 2 will be done by one of the proposed

models. We have experimented with several deep learning architectures. First, we designed a new CNN architecture from scratch specially designed for this application. We tried a lot of combinations for various layers to get the best results. Then we experimented with transfer learning using three existing deep learning architectures i.e., ResNet50, MobileNet, and Inception V3. All the experiments are done using Keras and Tensorflow in Google Collaboratory. Let us discuss their implementation in detail.

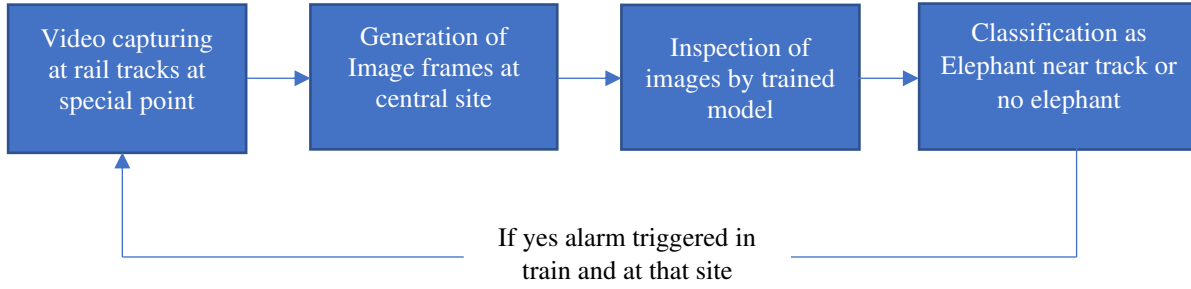


Fig. 2 Proposed model for Prevention of Train-Elephant collision

A. CNN Architecture

CNN stands for convolutional neural networks which aim at extracting features from images so that these can help to 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 for the identification and classification of 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, and others. 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 next 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. The number and type of layers if changed will cause a new architecture with a different feature set and outcome. Apart from that, a lot of hyperparameters are available to fine-tune the architecture. For feature extraction, we can use the different number of convolution layers with different parameters such as filter size, activation function, and

image size. For the pooling layer, stride can be selected as per requirement. For dense layers, the activation function such as 'Relu' or 'leakyrelu' can be selected 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, drop out and regularization can be used for tuning the architecture.

Details of Proposed CNN architecture: The proposed architecture has 5 convolutional layers and 5 pooling layers. Then the output is flattened, and a fully connected layer is employed for binary classification. Fig. 3 shows the output of convolution layers and pooling layers as a heat map. Fig. 4 shows the details of the proposed CNN model for classifying images of rail track with or without an elephant. Various parameter values used in the proposed CNN are shown. Data Augmentation is used so that model could become translation, rotation, and scaling invariant. A batch size of 32 is used for training as well as testing. 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 larger steps in the horizontal direction and can converge faster. Categorical_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. Drop out is used to limit the parameters in fully connected layers.

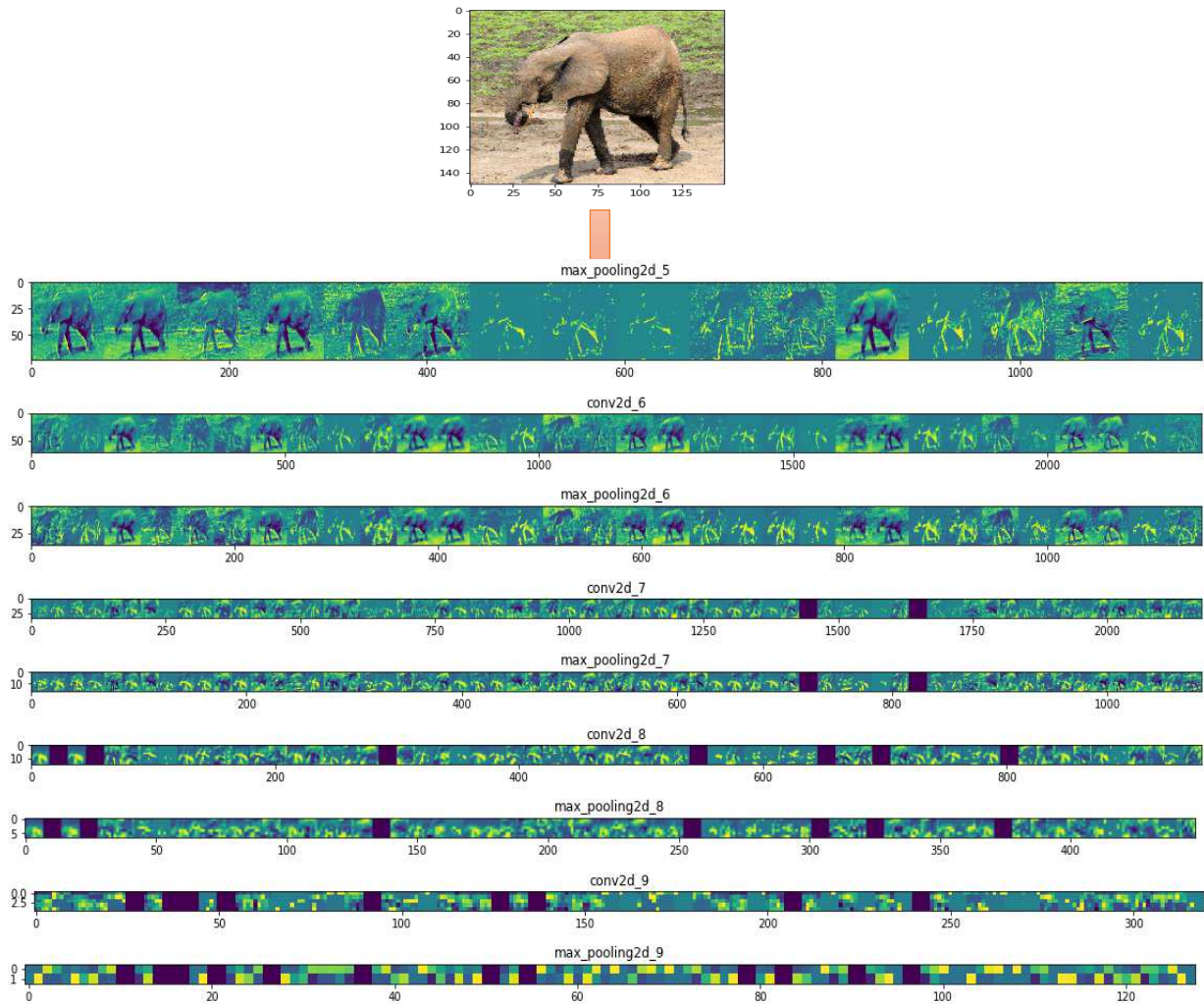


Fig. 3 Analysis of output of convolution layer of proposed CNN architecture

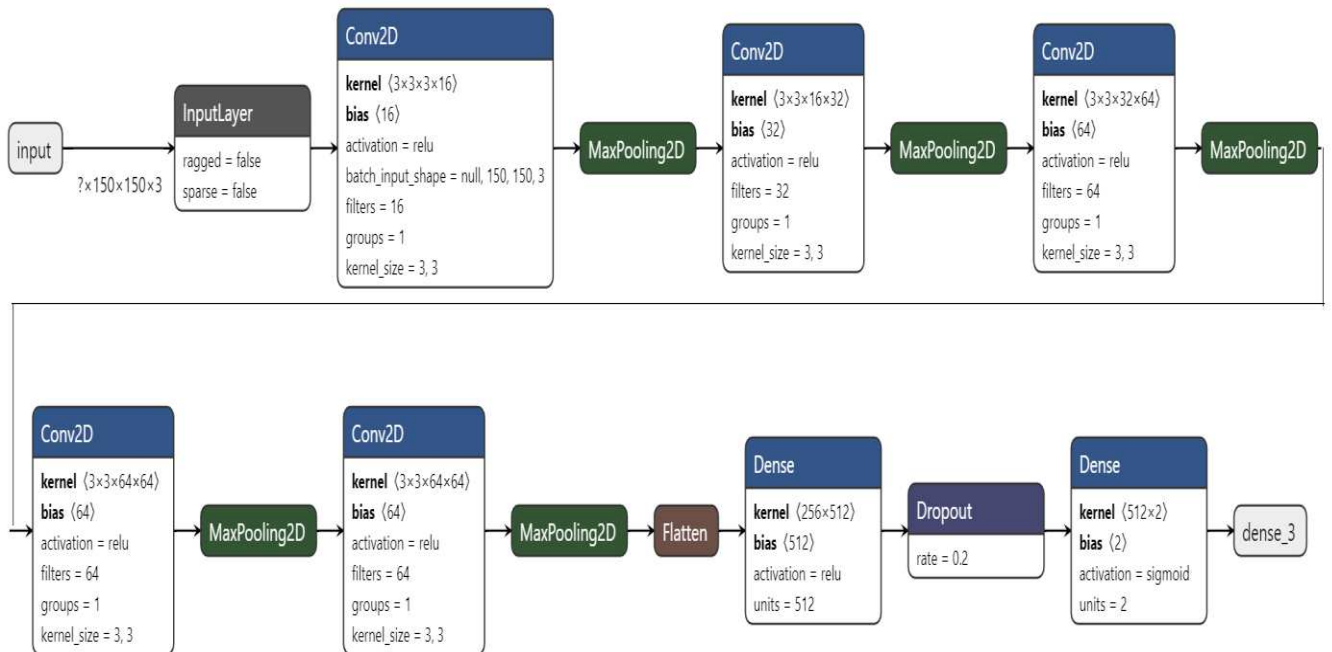


Fig 4. The detailed architecture of the proposed CNN

B. 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 realizing where pre-prepared models are utilized as the beginning point for the next model. The pre-trained model (trained using large datasets) is saved with the weights. 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. 5.

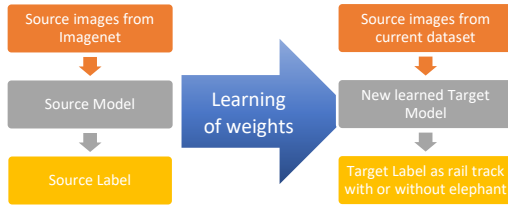


Fig. 5 Concept of Transfer Learning

The two basic methodologies for TL are as follows:

1. Develop Model Approach
2. Pre-trained Model Approach

There are three main steps in the pre-trained model approach. First, a pre-prepared source model is explored within accessible models. These pre-trained models are usually trained on huge datasets. A model trained on a similar dataset is usually selected from a pool of available models. Secondly, the selected model is reused. The pre-trained model is used as the beginning stage for a model on the second assignment of interest. This may include utilizing all or some parts of the model. Thirdly, the obtained model is finetuned and modified. Alternatively, the model may be adjusted or refined as per the undertaking of interest.

The **ImageNet** project is a large repository of a variety of images 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[37] i.e. Residual Networks is a neural network that is utilized as the foundation of numerous computer vision-based exercises. This model was the winner of the ImageNet challenge in 2015. Incredible advancement with ResNet permits to prepare top to bottom organizations with 150+ layers effectively. Earlier, ResNet usage for the exceptionally deep neural network was troublesome because of the issue of the vanishing of inclinations. ResNet then presented the idea of skip association. Now, ResNet skip associations are utilized in much more model structures like the Fully Convolutional Network (FCN) and U-Net. They are utilized to stream data from prior layers in the model to later layers. In these designs, they are utilized

to pass data from the downsampling layers to the upsampling layers. The ResNet-50 model consists of 5 stages each with a convolution and Identity block. Each convolution block has 3 convolution layers and each identity block also has 3 convolution layers. The ResNet-50 has over 23 million trainable parameters.

MobileNets[38] are a class of small, low-computation models that can be utilized for order, recognition, and other regular tasks that can be solved using convolutional neural networks. Considering their little size, these are viewed as incredible deep learning models to be utilized on cell phones. 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. Truly, MobileNets regularly aren't as precise as these other huge, asset substantial models are, however they still really perform quite well, with truly just a generally little decrease inexactness.

Inception/GoogleNet[39] Google devised a module called the inception module that approximates a sparse CNN with a normal dense construction. Since only a small number of neurons are effective, the width/number of the convolutional filters of a particular kernel size is kept small. Also, it uses convolutions of different sizes to capture details at varied scales (5X5, 3X3, 1X1). It exploits the fact that most of the 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 have a connection with all the 512 input channels. Another salient point about the module is that it has bottleneck layer 1X1 convolutions. It helps in the massive reduction of the computation requirement.

V. RESULT AND DISCUSSION

As mentioned in the methodology, four different CNN models have 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 has the advantage that it is a lightweight model with only seven layers and about 2 lakhs parameters. Fewer parameters make the proposed model computationally efficient. Transfer learning-based models have performed better as they have more layers and more parameters. Resnet50 and MobileNet performed better with an accuracy of 99.81%. Inception Net has performed best in binary classification. It achieved an accuracy of 99.91%. The inception Model has a relatively less number of layers and parameters and hence is suited best for the current problem.

Fig. 6 shows the confusion matrix for the Test dataset for 1056 images. 505 images contain elephant(s) in frames and 551 images are without an elephant. Fig 6 a) shows the confusion matrix for the proposed lightweight CNN. Fig. 6 b, c, d shows the confusion matrix for transfer learning

models. For the proposed CNN only 5 images out of 1056 are categorized wrongly. The performance of TL models is better. Only 2 images are wrongly classified 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.

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
Mobile net	28	4 million	99.81%	1.0	1.0	1.0
Inception V3	22	5 million	99.91%	1.0	1.0	1.0

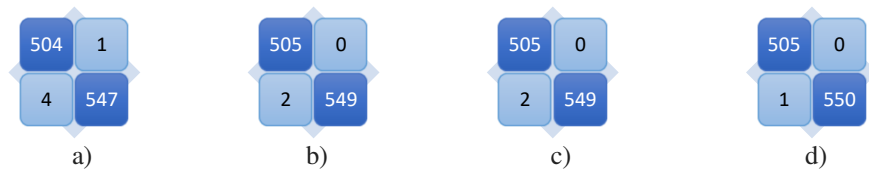


Fig. 6 Confusion Matrix a) proposed CNN, b) Resnet50 c) MobileNet d) Inception V3

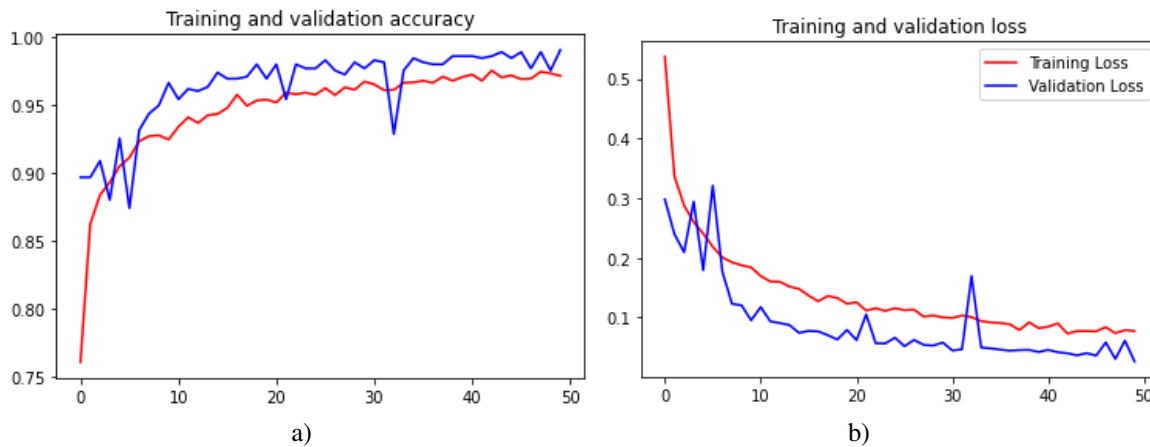


Fig. 7 Training and validation a) accuracy and b) Loss for proposed CNN model

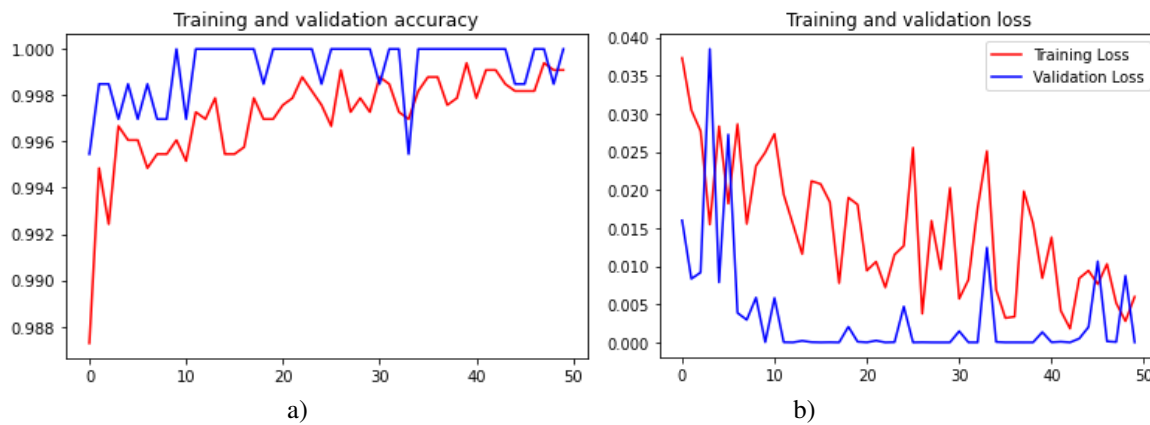


Fig. 8 Training and validation a) accuracy and b) Loss for the Inception V3 model

Fig. 7a, 7b shows the training and validation accuracy and loss for the proposed CNN, and Fig. 8a, 8b shows the same for the best transfer learning model i.e. Inception model. The trend in the graph shows that validation accuracy and loss has followed the training accuracy and loss trend.

Table 4. Comparison with the existing state of art

Method	Dataset size	FPR	Accuracy
Zeppelzauer et al., 2015[25]	715	2.5%	91.7%
Ravikumar et al., 2020[31]	1792	3.1%	92.7%
Proposed CNN	4305	0.18%	99.53%
Proposed Inception v3	4305	0	99.91%

Table 4. shows the comparison of the proposed model with similar work. Zeppelzauer et al. [27] proposed a multi-modal early warning system for the detection of elephants in wild video recordings. Elephants are identified based on the color model of their body. SVM classifier is 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., [33] proposed MobileNet architecture along with the single-shot detection algorithm for the detection of elephants in wild videos. This CNN model achieved an accuracy of 92.7%. Our proposed CNN and tuned

Inception model outperformed the existing models with better accuracy and False Positive Ratio. Therefore, the proposed model is better than the existing model.

TESTING BEYOND DATASET:

When the testing is done for images with an elephant on track and rail track with no one on track then accuracy obtained is almost 100%. But we tried 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. Fig. 9 shows the results obtained by the Inception V3 model for typical 40 real-time images found on the Internet. It had been 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 a real situation. For few images are shown in Row 1 with big animals are misclassified as an image with an elephant. The poor resolution of these images is also one of the constraints. This can certainly be improved with a larger, ideal dataset which is not available as such.

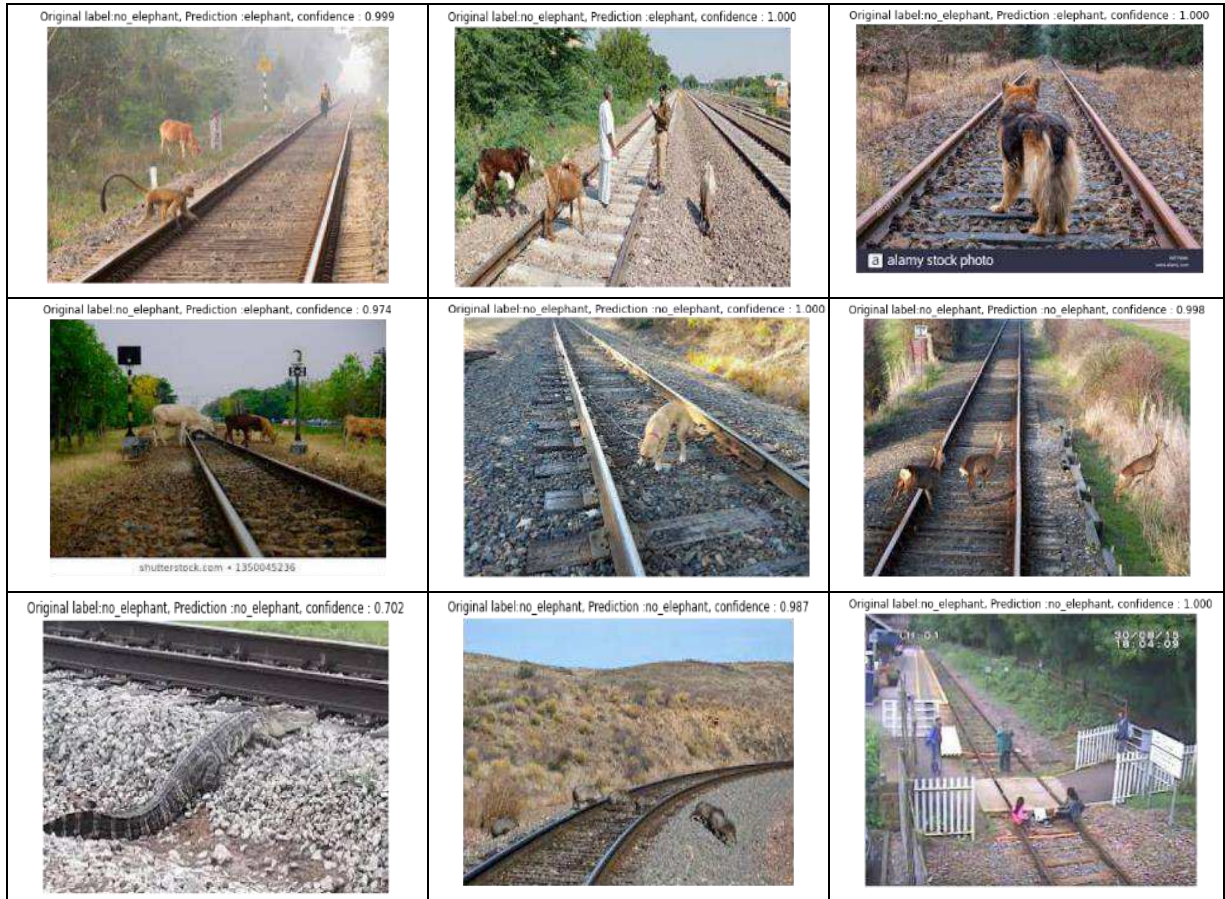




Fig. 9 Result of Inception V3 model on unknown 40 images from the internet

VI. CONCLUSION

As human activities had led to intervention in wildlife, its consequences are visible in terms of animal extinction. Nowadays a lot of efforts are being made for animal conservation and advanced technology can certainly help in this regard. Artificial Intelligence 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. Four different models based on CNN and TL are proposed and compared for w.r.t their effort and efficiency. The Inception v3 model has performed best for this application because of its high accuracy and zero true negative rates. The model can be used for generating alarm on-site and in a train near the track for warning and hence saving elephant life. In the future, a detailed dataset can be prepared and trained so that the other type of animals may not be misclassified as elephants and hence can lead to zero false-positive cases.

Author Contributions:

All authors contributed to the review of literature, the study of methodologies, design and implementation of the proposed method. Material preparation, literature review, analysis and implementation were performed by Neeraj Mohan, Madhavi Karanam, Krishna Chythanya and Surbhi Gupta, respectively. The first draft of the manuscript was written by Surbhi Gupta and all authors commented on previous versions of the manuscript. All authors have thoroughly read, commented and approved the final manuscript.

Conflicts of interest/Competing interests: There is no conflict of interest by any of the authors.

References:

- Langbein, J., 2011. Monitoring reported deer road casualties and related accidents in England to 2010. Research Report 2011/3. The Deer Initiative, Wrexham, UK.
- 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.
- Chythanya K., Madhavi K., and Ramesh G., 2020, A Machine Learning Enabled IoT Device to Combat Elephant Mortality on Railway Tracks. Springer Proceedings of 2nd International Conference on Innovative Data Communication Technologies and Applications (ICIDCA 2020)-Sept.2020. In press.
- Sharkawy, A.N., Koustoumpardis, P.N. and Aspragathos, N., 2020. Human-robot collisions detection for safe human-robot interaction using one multi-input-output neural network. *Soft Computing*, 24(9), pp.6687-6719.
- Praczyk, T., 2020. Neural collision avoidance system for biomimetic autonomous underwater vehicle. *Soft computing*, 24(2), pp.1315-1333.
- Koik, B.T. and Ibrahim, H., 2012. A literature survey on animal detection methods in digital images. *International Journal of Future Computer and Communication*, 1(1), p.24.
- Tanwar M., Shekhawat N.S., Panwar S., 2017, A Survey on Algorithms on Animal Detection-International Journal on Future Revolution in Computer Science & Communication Engineering ISSN: 2454-4248. Volume: 3 Issue: 6, pp. 33 – 35.
- Tweed, D. and Calway, A., 2002, August. Tracking multiple animals in wildlife footage. In *Object recognition supported by user interaction for service robots*, Vol. 2, pp. 24-27, IEEE.
- Ramanan, D. and Forsyth, D.A., 2003, October. Using temporal coherence to build models of animals. pp. 338. IEEE.
- Hannuna, S.L., Campbell, N.W. and Gibson, D.P., 2005. Segmenting quadruped gait patterns from wildlife video. 2005.
- Burghardt, T. and Čalić, J., 2006. Analysing animal behaviour in wildlife videos using face detection and tracking. *IEE Proceedings-Vision, Image and Signal Processing*, 153(3), pp.305-312.
- Farah, R., Langlois, J.P. and Bilodeau, G.A., 2011, September. Rat: Robust animal tracking. In *2011 IEEE International Symposium on Robotic and Sensors Environments (ROSE)* (pp. 65-70). IEEE.
- Mammeri, A., Zhou, D., Boukerche, A. and Almulla, M., 2014, June. An efficient animal detection system for smart cars using cascaded classifiers. In *2014 IEEE*

International Conference on Communications (ICC) (pp. 1854-1859). IEEE.

14. Sharma, S.U. and Shah, D.J., 2016. A practical animal detection and collision avoidance system using computer vision technique. *IEEE Access*, 5, pp.347-358.
15. Norouzzadeh, M.S., Nguyen, A., Kosmala, M., Swanson, A., Palmer, M.S., Packer, C. and Clune, J., 2018. Automatically identifying, counting, and describing wild animals in camera-trap images with deep learning. *Proceedings of the National Academy of Sciences*, 115(25), pp. E5716-E5725.
16. Raja, M.A.A., Ramya, M.K., Kousalya, M.B. and Jeeva, M.S., 2018. Prevention of wild animals from accidents using image detection and edge algorithm. *PREVENTION*, 5(11).
17. Devost, E., Lai, S., Casajus, N. and Berteaux, D., 2019. Fox Mask: a new automated tool for animal detection in camera trap images. *BioRxiv*, p.640037.
18. Zotin, A.G. and Proskurin, A.V., 2019. Animal detection using a series of images under complex shooting conditions. *International Archives of the Photogrammetry, Remote Sensing & Spatial Information Sciences*.
19. Backs, J.A.J., Nychka, J.A. and Clair, C.S., 2017. Warning systems triggered by trains could reduce collisions with wildlife. *Ecological Engineering*, 106, pp.563-569.
20. Bíl, M., Andrášik, R., Duľa, M. and Sedoník, J., 2019. On reliable identification of factors influencing wildlife-vehicle collisions along roads. *Journal of environmental management*, 237, pp.297-304.
21. Jayakumar R., Swaminathan R., Harikumar, S., Banupriya N., Saranya S., 2020. Animal detection using deep learning algorithm. *J Crit Rev*, 7(1), pp.434-439.
22. Venkataraman, A.B., Saandeep, R., Baskaran, N., Roy, M., Madhivanan, A. and Sukumar, R., 2005. Using satellite telemetry to mitigate elephant-human conflict: An experiment in northern West Bengal, India. *Current Science*, pp.1827-1831.
23. Ardovini, A., Cinque, L. and Sangineto, E., 2008. Identifying elephant photos by multi-curve matching. *Pattern Recognition*, 41(6), pp.1867-1877.
24. Vermeulen, C., Lejeune, P., Lisein, J., Sawadogo, P. and Bouché, P., 2013. Unmanned aerial survey of elephants. *PloS one*, 8(2), p.e54700.
25. Zeppelzauer, M., 2013. Automated detection of elephants in wildlife video. *EURASIP journal on image and video processing*, 2013(1), p.46.
26. Sugumar, S.J. and Jayaparthathy, R., 2014. An improved real time image detection system for elephant intrusion along the forest border areas. *The Scientific World Journal*, 2014.
27. Zeppelzauer, M. and Stoeger, A.S., 2015. Establishing the fundamentals for an elephant early warning and monitoring system. *BMC research notes*, 8(1), p.409.
28. Shukla, P., Dua, I., Raman, B. and 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).
29. Mandal, R.K. and Bhutia, D.D., 2018. A Proposed Artificial Neural Network (ANN) Model Using Geophone Sensors to Detect Elephants Near the Railway Tracks. In *Advanced Computational and Communication Paradigms* (pp. 1-6). Springer, Singapore.
30. Marais, J.C., 2018. *Automated elephant detection and classification from aerial infrared and colour images using deep learning* (Doctoral dissertation, Stellenbosch: Stellenbosch University).
31. Dhanaraj, J.S.A. and Kumar Sangaiah, A., 2018. Elephant detection using boundary sense deep learning (BSDL) architecture. *Journal of Experimental & Theoretical Artificial Intelligence*, pp.1-16.
32. Kumar S., Baline H. V., Sivakumar T., and Potluri V. P., 2019, Detection of wild elephants using image processing on raspberry PI3- *International Journal of Computer Science and Mobile Computing*, Vol.8 Issue.2, February- 2019, pp. 104-115.
33. Ravikumar, S., Vinod, D., Ramesh, G., Pulari, S.R. and Mathi, S., 2020. A layered approach to detect elephants in live surveillance video streams using convolution neural networks. *J. Intell. Fuzzy Syst.*, 38(5), pp.6291-6298.
34. Naude, J. and 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).
35. Korschens, M. and 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).
36. Zendel, O., Murschitz, M., Zeilinger, M., Steininger, D., Abbasi, S. and Beleznaï, 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).
37. <https://neurohive.io/en/popular-networks/resnet/>
38. Howard, A.G., Zhu, M., Chen, B., Kalenichenko, D., Wang, W., Weyand, T., Andreetto, M. and Adam, H., 2017. Mobilenets: Efficient convolutional neural networks for mobile vision applications. *arXiv preprint arXiv:1704.04861*.
39. Szegedy, C., Vanhoucke, V., Ioffe, S., Shlens, J. and 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).

Figures



Figure 1

Sample images used for training and testing

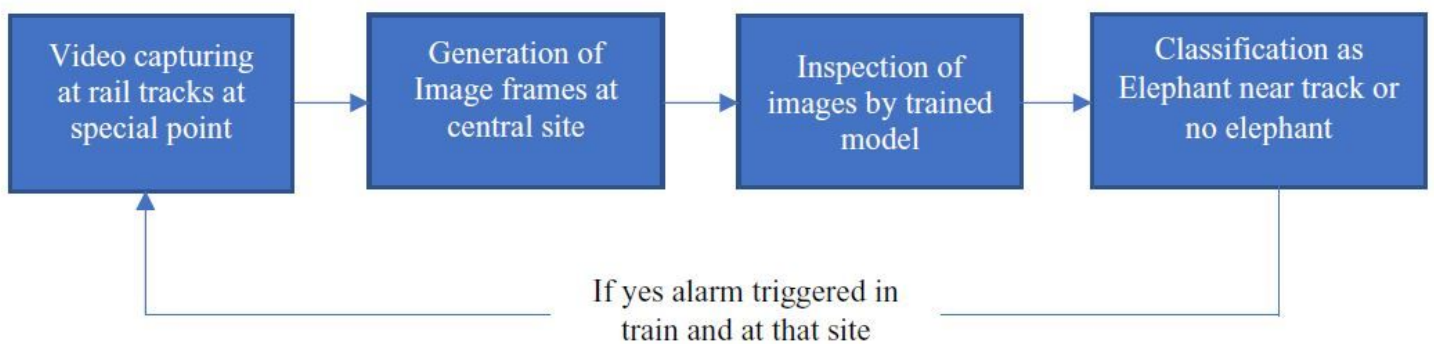


Figure 2

Proposed model for Prevention of Train-Elephant collision

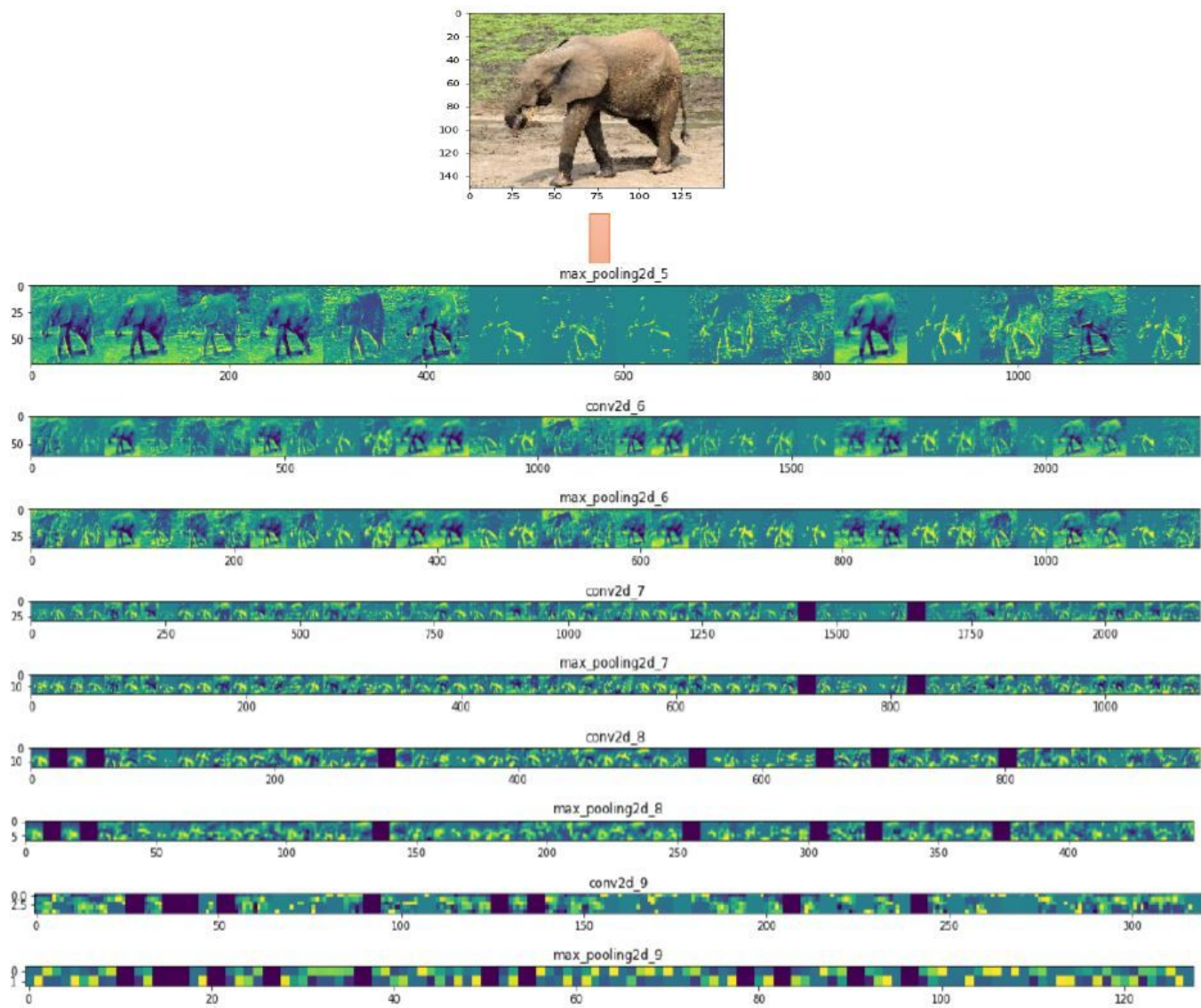


Figure 3

Analysis of output of convolution layer of proposed CNN architecture

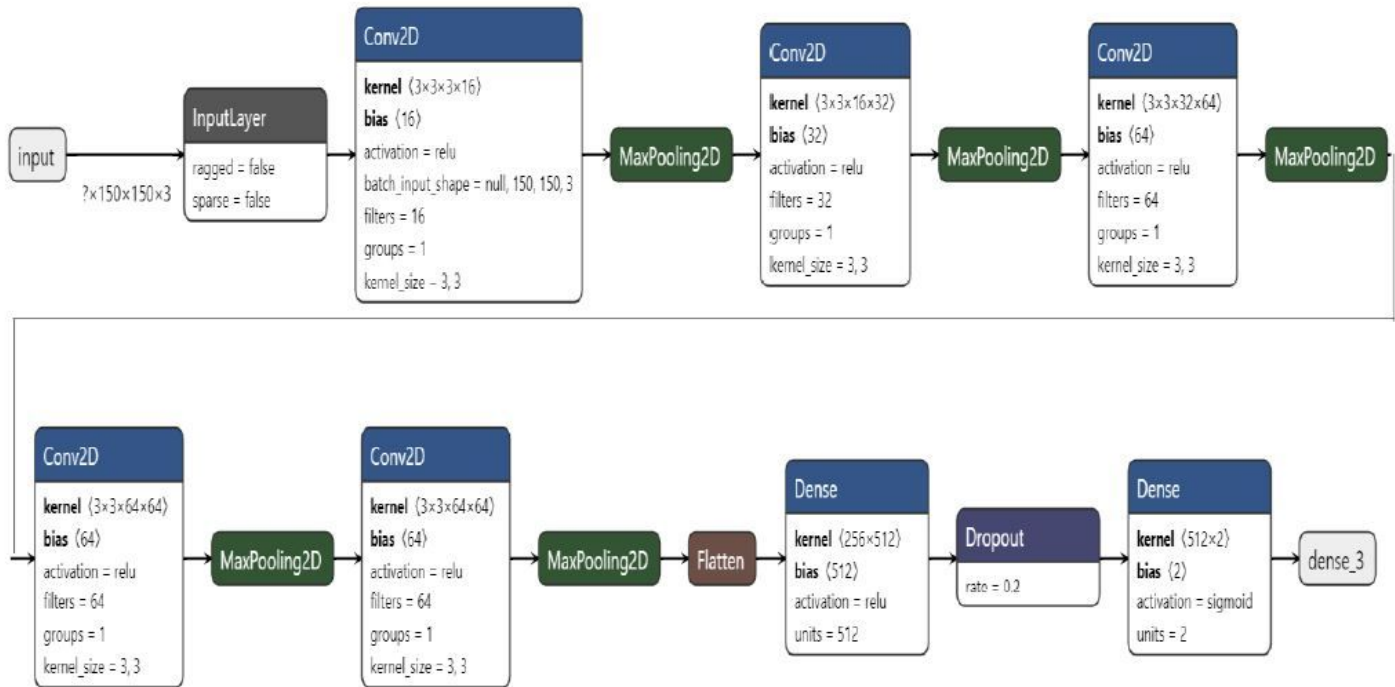


Figure 4

The detailed architecture of the proposed CNN

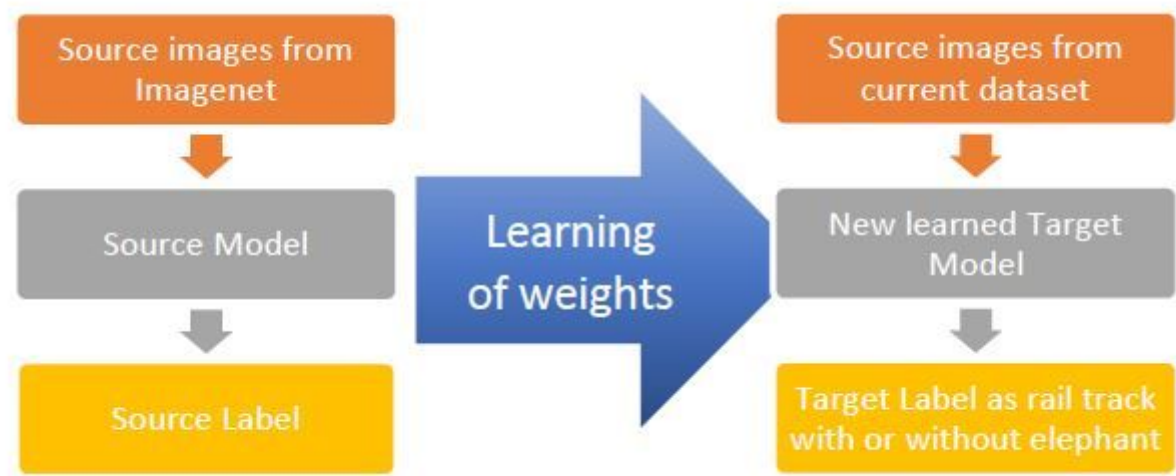


Figure 5

Concept of Transfer Learning



Figure 6

Confusion Matrix a) proposed CNN, b) Resnet50 c) MobileNet d) Inception V3

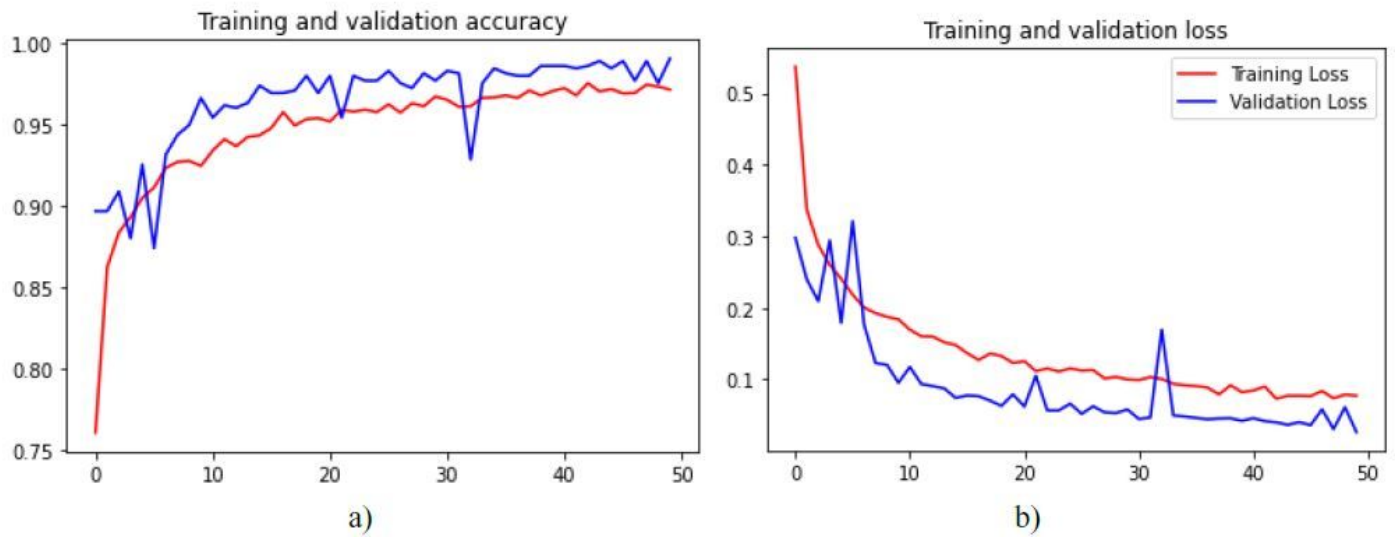


Figure 7

Training and validation a) accuracy and b) Loss for proposed CNN model

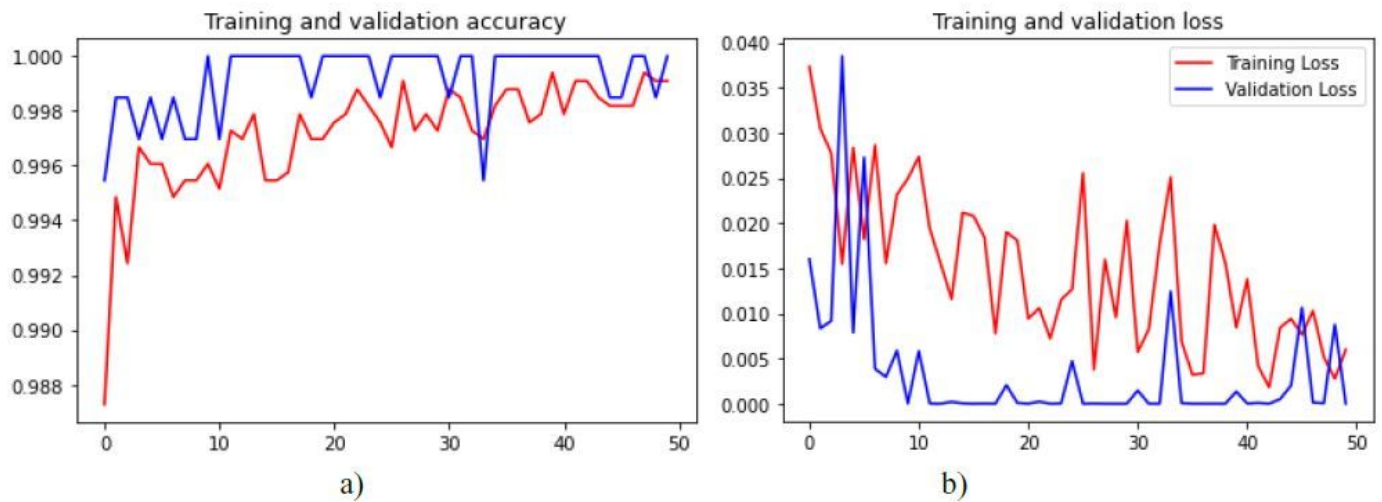


Figure 8

Training and validation a) accuracy and b) Loss for the Inception V3 model

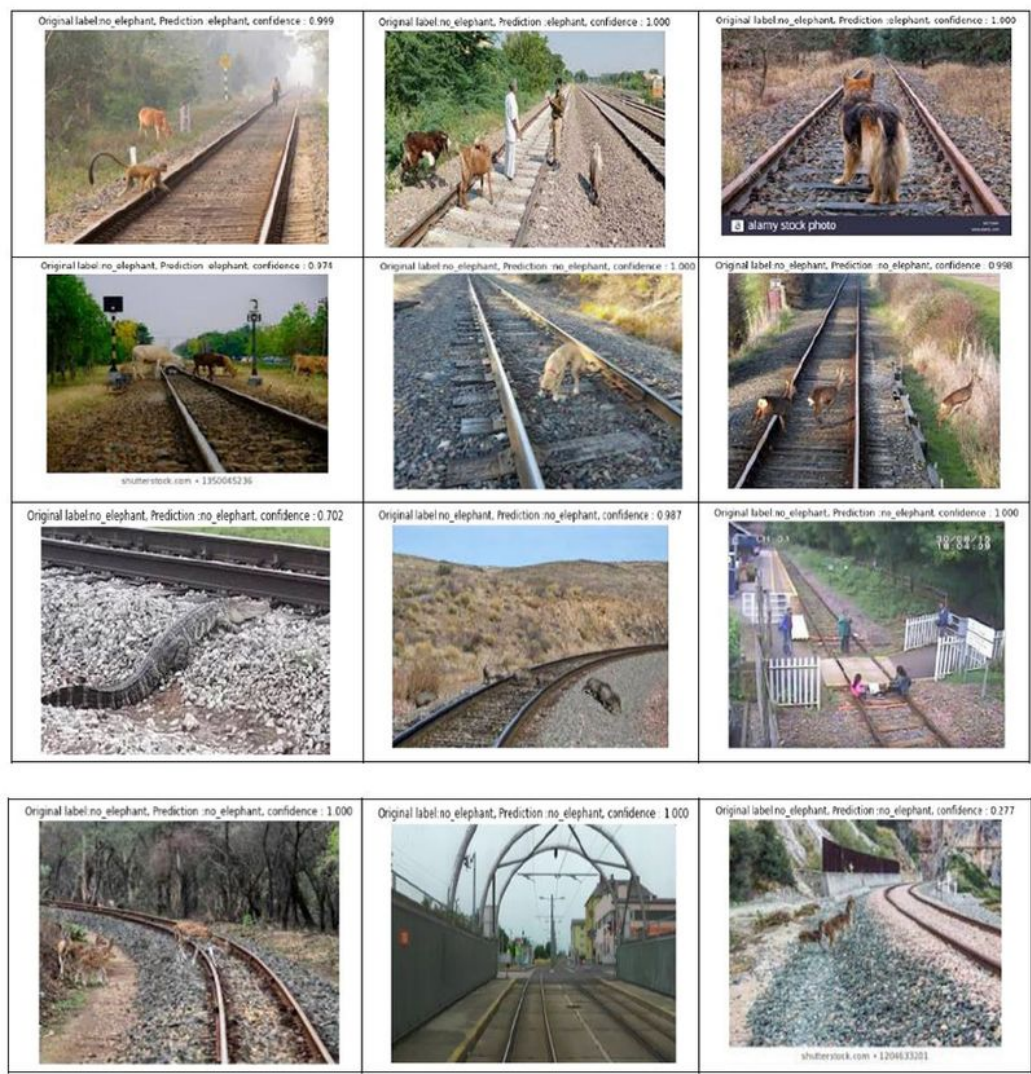


Figure 9

Result of Inception V3 model on unknown 40 images from the internet

Supplementary Files

This is a list of supplementary files associated with this preprint. Click to download.

- [CNNmodel.pdf](#)