On-Road Deer Detection for Advanced Driver Assistance using Convolutional Neural Network

W Jino Hans[1], N Venkateswaran[3]
ECE,
SSN College of Engineering
Chennai, India

V Sherlin Solomi[2]
ECE,
Hindustan Institute of Technology and Science
Chennai, India

***Abstract*—Animal-vehicle collision (AVC) is a major concern in road safety that affects human life, properties, and wildlife. Most of the collisions happen with large animals especially deer that enters the road suddenly. Furthermore, the threat is even more alarming in poor visibility conditions such as night-time, fog, rain, etc. Therefore, it is vital to detect the presence of deer on roadways to mitigate the severity of deer-vehicle collision (DVC). This paper presents an efficient methodology to detect deer on roadways both during the day and night-time conditions using deep learning framework. A two-class CNN model differentiating a deer from its background is developed. The background will have a few classes of objects such as motorcycles, cars, and trees which are frequently encountered on roadways. A self-constructed dataset with both RGB and thermal images is used to train the CNN model. Sliding window technique is used to localize the spatial region of deer in an image. The performance of the proposed CNN model is compared with state-of-the art classifiers and pre-trained CNN models and the results validate its effectiveness.**

Keywords*—Computer vision; animal detection; deep learning; Animal Vehicle Collision (AVC)

I. Introduction

Intelligent transportation system (ITS) is a technology, application or a platform that improves the quality of transportation. The main aim of ITS is to serve the public good by leveraging technology to maximize safety, mobility, and environmental performance [1]. It plays a significant role in crucial decision-making tasks to improve the overall operation of transportation system. A major subset of ITS is advanced driver assistance systems (ADAS) which focuses on the safety of drivers and vulnerable road users [2]. The word "vulnerable" is used to describe people who are disproportionately represented in road accidents especially children, elderly and physically challenged. In addition, animals can also be categorized under vulnerable category due to its poor sensory perception and its unpredictable movement patterns.

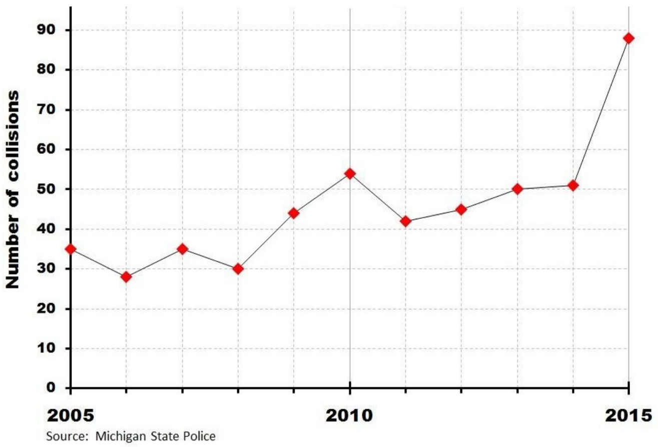
In recent years, pedestrian detection have garnered a special interest among researchers. Numerous algorithms and methodologies to efficiently detect and track pedestrians have been proposed for better decision-making to avoid road accidents [3] [4]. However, considerably lesser contributions have been reported in literature to detect and track animals. Animal-vehicle collision (AVC) is a serious problem that affects human safety, property and wildlife. The number of these collisions has increased substantially over the last decades [5]. An often ignored aspect of road accidents involves human injuries and deaths due to road collisions with animals [6].

AVC is a challenging problem across the globe. J.M. Conn et al. reported that in the US, more than 1.5 million accidents happen due to animal collisions every year which results in approximately 200 human deaths and 29,000 injuries. Furthermore, a property damage worth 1.1 billion dollars has also been reported in the US [5]. A similar statistics is seen in other nations as well. For instance, Meister et al. shows that in Canada approximately 50% of road users report hitting a wildlife [7]. European countries also have witnessed more than 500000 collisions with large animals which caused more than 300 fatalities and 30000 injuries [8] [9]. Moreover, a report by the Indian ministry of road transport and highways showed that due to widening of national highways through forested areas and wildlife corridors, wild animals crossing national highways caused approximately 8000 accidents between 2006 and 2012 [10] [11]. Wild animals such as elephants, deer, leopards and tigers often cross highways in forested areas, resulting in road accidents. Furthermore, it is also reported that road-kill of wild animals had become a significant threat to wildlife population.

Despite many species of large animals have been reported for fatal AVCs, it is observed that almost 77 percentage of accidents occur with deer [12] [13]. Fig. 1 depicts a few examples for sudden crossings of deer on roadways. It is estimated that in the US more than 1 million accidents occurs with deer. Similar evidence exists for Europe as well with 0.5 million accidents involving a deer [14]. Furthermore, the number of collisions with deer is increasing year-on-year and hence deer-vehicle collision (DVC) is a serious threat to road safety. Fig. 2 depicts the statistics for road accidents caused due to wildlife provided by the annual Michigan Police Report at Ann Arbor, Michigan, United States of America.

Fig. 1. A few examples for sudden deer crossing on roadways

It is observed that a specific activity pattern of deer has a strong influence over DVCs. For instance, peak fatal collisions with deer happens during the month of June and November which coincides with the breeding and migration season of deer respectively [15]. To mitigate the severity of DVC, several systems and models have been proposed over the past two decades [16] [17].



Source: Michigan State Police

Fig. 2. A Statistics on DVC between 2010 and 2015 in Michigan, United States (Source: Michigan Police Report)

DVC mitigation systems can be classified into two main group viz. passive and active methods. In passive methods, detterence strategies such as the use of ultrasonic whistles [Hornet V120], high intensity lights, animal reflectors, electric fences, and roadside refractors are used to warn and keep the deer away from roadways [16]. Despite the fact that these techniques are popular, they seem to be quite inefficient and outdated as the animal will get accustomed to these systems [18]. On the contrary, active methods are based on animal detection which involves techniques and strategies to detect deer in the vicinity of the vehicle [18]. The active methods can be either a sensor based approach (ultrasonic devices) or camera based approach (computer vision). Despite the fact that, ultrasonic devices are used widely to capture any obstacle or animal within its range, it requires a clear line of sight to establish beam connection. Furthermore, these systems are prone to give false alarms when encountered with other obstacles such as small animals, vehicles crossing in the other lane or the air movement. Moreover, Mammeri et al. suggests that camera-based techniques are more efficient and reliable to detect deer rather than sensor-based approaches [19]. It is due to the fact that camera-based systems can efficiently visualize the regions under investigation to detect the presence of deer. However, the limitation of these camera-based system is that its efficiency is dependent on the field of view (FOV) of the camera. Often, the FOV of the camera falls within the road. Deer that are seen within the road will be under the FOV and hence will have a high chance of being detected and those outside will have a risk of being missed. Furthermore, detecting deer during night-time and on a curve-roads is also a challenging task.

The main focus of this paper is to propose an efficient technique to detect deer on roadways using sophisticated computer-vision techniques. It is quite similar to pedestrian detection and hence it is possible to adapt the techniques and methodologies employed in pedestrian detection for deer detection as well. Most classical pedestrian detection algorithm primarily have two steps viz. feature extraction and classification [20]. Texture features such as Haar-like features [21], Local Binary Patterns (LBP) features [22], [23] and gradient features such as Histogram of Gradients (HOG) [24] are widely used in pedestrian detection algorithms. Moreover, state-of-the-art machine learning techniques such as AdaBoost and SVM are used as a classifier in these algorithms [20]. However, most of these techniques does not provide impressive results when used directly for animal detection as it requires a particular pose of the animal in the dataset [19].

In recent years, due to the rapid development and evolution in Graphics Processing Unit (GPU), the use of deep learning systems have shown a significant leap. Deep learning techniques especially Convolutional Neural Network (CNN) have witnessed a huge success in many computer-vision tasks more particularly on recognition and classification applications [25] [26] [27]. Owing to the success of CNN in various image analysis tasks, it is believed that CNNs can be an ideal solution for deer detection problem as well. In this paper, we explore the use of CNN for the detection of deer on roadways both during day and night-time. A two class CNN classifier is trained to classify an image as 'deer' or 'not a deer'. The CNN classifier is trained using a self-constructed dataset consisting of both RGB and thermal images of deer and its background. The background class will have any of the three sub-classes viz. motorcycles, cars and trees. Moreover, the silhouettes of the images are used as a feature to train the CNN model. It is believed that training the CNN with the silhouette of deer will improve the overall detection accuracy of the model particularly in night-time as the silhouette of deer in both RGB and thermal image remains quite similar.

The main contributions of this paper are as follows:

- A deep learning approach to detect deer on roadways during both day and night-time conditions is proposed.
- A large self-constructed dataset containing both RGB and thermal images of deer with different poses is created.
- The spatial region of deer in an image is localized using sliding window technique.

The rest of the paper is organized as follows: Section II review a few related work in animal detection whereas, Section III presents the proposed methodology to detect deer on roadways. Experimental results and evaluation are reported in Section IV and Section V concludes the paper.

II. Background and Related Work

A few state-of-the-art detection algorithms that are quite successful are considered for the study. First of all, we study and compare texture feature based detection algorithms. T. Burghardt et al. detected the faces of lion using Haar-like features that are used primarily to detect human faces [28]. The features are trained by a cascaded AdaBoost classifier. Furthermore, the authors have extracted these features from a color map and claim it to be more robust to shadows and illumination changes. Despite the fact that this method is effective in detecting Lion-faces, the algorithm is successful only when the anterior view of the animal is within the FOV of camera. Another successful technique that is quite popular is the gradient feature based detection algorithm. Gradient features such as Scale-invariant Feature Transform (SIFT) or Histogram of Oriented Gradient (HOG) will describe the object's edge and contour appropriately [24]. SVM classifiers are successfully used with HOG features to detect faces, pedestrians [29]. D. Ramanan et al. proposed a animal detection algorithm that effectively builds a visual model of the animal from videos [30]. It is based on an assumption that an animal can be well represented as a combination of small regions with significant body parts in it. The authors have employed three features viz. Histogram of Textons, intensity-normalized patch pixel values and scale-invariant feature transform (SIFT) descriptors to identify the animal with the use of three different classifiers, namely, K-logistic regression, SVM and K-Nearest neighbor (KNN). It is observed that KNN outperforms SVM and K-logistic regression, if K value is chosen as 1. Despite this system achieves better results in terms of classification error, the algorithm lacks speed as it uses SIFT which is a local descriptor as compared with a global descriptor such as HOG. Moreover, its application is limited to animals with lateral view alone and hence pose a few limitation in natural scenario. A similar approach is adopted to detect penguins by T. Burghardt et al. [31]. AdaBoost classifier is used to train the features of penguins for detecting the presence of unique black spot in the chest of adult penguins. However, it has not yielded satisfactory results when the penguin has a different feather pattern or the anterior view of penguins are not available.

Z. Debao et al. proposed a technique to detect deer by finding the contour of the animal in a small segmented region [32]. Images are segmented into small regions based on its intensity levels and are resized to fit the contour of the animal. Furthermore, HOG features are extracted from the segmented ROI instead of the entire image and a linear SVM classifier is used for detection. This method achieves better detection accuracy for animals in a close proximity. However, the detection is not very accurate and produces unfaithful results if the animal is far away. Zhang et al. proposed an effective method to detect head of animals in images by extracting Haar features in four channels [33]. These Haar-like features named as Histogram of oriented gradients (HOG) are used to capture the local shape and variation of textures in an image. These features are effectively used to detect animal head by using a deformable detection technique. Despite this technique produces more promising results, it works well only for images with animal heads in it. Apart from texture and shapes, color features are also used to detect animal. M. Zeppelzauer et al. proposed a technique based on color segmentation [34]. Each and every animal will have a dominant color and the regions in the image with that particular color hue are segmented using mean-shift clustering algorithm. A color model for a particular animal is learned from a set of training images. Unfortunately, this method is well suited for day-time conditions and cannot be used in night-time. Khorrami et al. used Principal Component Analysis (PCA) to detect different species of animals by reducing the data dimensionality [35]. The above mentioned works indicate that most of the conventional features such as texture, gradient and color features can be used for animal detection task. Owing to the success of deep learning algorithms in image classification tasks, we explore the use of CNN for effective deer detection.

III. Proposed Deer Detection Methodology

The overview of the proposed CNN-based deer detection methodology is illustrated in Fig. 3. The proposed deer detection methodology consists of training phase (performed offline), during which an efficient model based on CNN is trained from a collection of positive and negative dataset. A dash-board camera installed inside the vehicle is used to capture the frames in front of the vehicle. The frames are then given as input to the CNN classifier to detect the presence of deer. The robustness of the model depends mainly on the dataset used in training. Hence, a large dataset containing both positive and negative images is created to train the model.

A. Dataset Creation

A self-constructed dataset with images of both deer (positive dataset) and its general surroundings (negative dataset) is created. Moreover, to efficiently train the model for night-time vision, images taken during night is also collected. It is a challenging task to create an extensive dataset with deer on roadways in different poses and time. Moreover, there is no public benchmarked dataset available exclusive for deer detection.

The dataset used in this research is primarily self acquired by collecting image frames from videos recorded by a dashboard camera installed in the car as depicted in Fig. 4. Approximately 10 hours of videos was recorded in different roadways including urban, rural and forested areas near Chennai city, India. Moreover, the videos are recorded in different weather conditions as well to make the training more robust. The captured videos are sampled at a rate of 1:4 to extract image frames without overlapping. Furthermore, FLIR E40 thermal camera is used to capture images of deer in night conditions. We have collected 2150 thermal images of chital deer and the blackbuck using the thermal camera. Apart from this, a few images containing deer, motorcycles and cars are collected from benchmarked public datasets such as CIFAR 100 and Caltech 256. In addition, we have incorporated a few data augmentation technique such as translation, cropping, flipping and also changed the lighting condition. The images are translated to four corners and thus obtained four additional augmented images. Cropping is done by 1 pixel in all four directions with respect to the co-ordinates of the deer area. Moreover, the images are flipped to left and right and finally the lighting conditions. This is achieved by adding Gaussian noise in the image. These augmented images are used along with the original images for training. The augmentation technique is applied only for positive dataset as the negative dataset is sufficiently large.

1) Positive Dataset: A positive dataset is created from the acquired images by selecting only those images having deer in it. The images in the positive dataset are classified based on the shape of deer due to its diverse postures. Despite deer can have different postures, only a few postures that are commonly encountered in roadways such as anterior, posterior and lateral view of the animal is considered. Furthermore, in lateral view the shape of deer will have two prominent shapes as the deer can either face to the right or to the left. Hence, more images with 'deer facing right' and 'deer facing left' are collected. These two shapes are very common and are

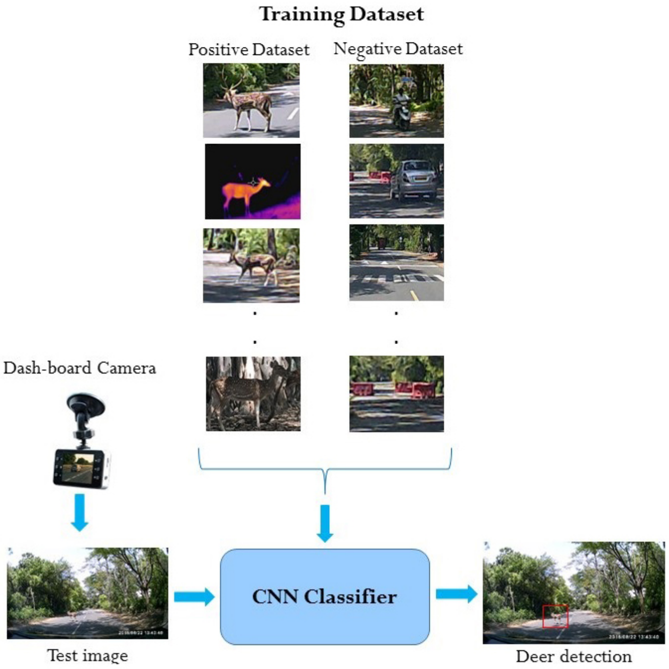


Fig. 3. An overview of the proposed CNN-based deer detection methodology

Fig. 4. Dashboard camera installed in a car to capture test videos

frequently encountered on roadways compared with anterior and posterior views. Fig. 5 depicts a few samples of positive dataset with deer in the above mentioned postures. A total of 5050 images which includes 2150 thermal and 2900 RGB images with the four above mentioned postures are collected for training the model.

2) Negative Dataset: A negative dataset improves the detection accuracy by decreasing the number of false positive detection. The negative dataset is created by collecting images of the objects that are frequently encountered in the deer detection scenario. A few images in the negative dataset is shown in Fig. 6. Any images that have objects with similar shape of a deer are excluded from the negative dataset to reduce the possibility of false positive detection. A few objects that are frequently encountered on roadways such as motorcycles, cars, pedestrians, sign boards, pavements, traffic lights, etc. are considered as the negative dataset. Moreover, these images are collected in different illumination and background conditions. We have collected 11450 images with the background for training the model.

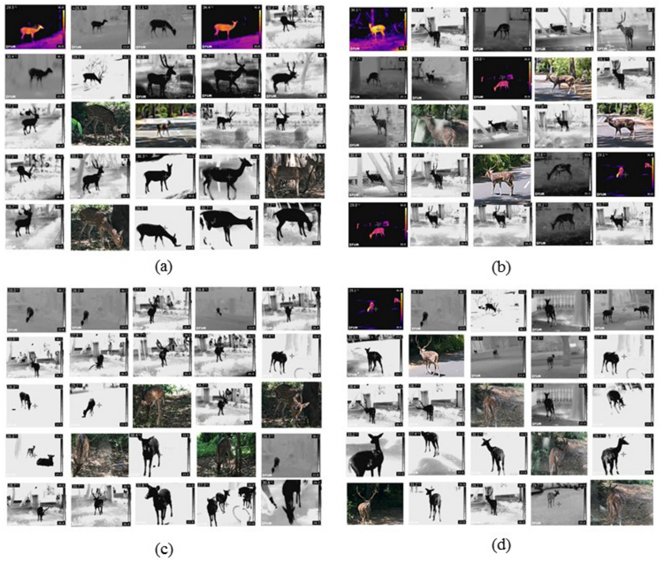


Fig. 5. Sample images in Deer dataset with different poses (a) Lateral view: Deer facing right (b) Lateral view: Deer facing left (c) anterior view (d) posterior view

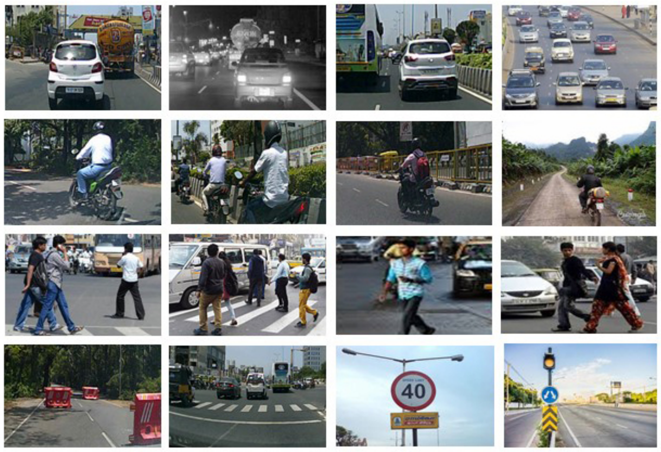


Fig. 6. Sample images of negative dataset

B. Pre-Processing

It is vital to perform a few pre-processing steps on the training images such that it can be used effectively for further processing. The pre-processing steps that are carried out in this work are discussed in this section.

1) Standardization of images: The first step in image pre-processing is to standardize the images in the dataset such that all images have a uniform aspect ratio and size. The training images are collected from different sources and hence will have different sizes and shapes. An important step in pre-processing is to obtain a standard aspect ratio (width:height) for the training images. The body ratio of a deer will be rectangular as the length of the animal is higher than its height. For instance an Indian chital deer will stand 0.7-0.9 m to its shoulder and its head to body length will be approximately 1.5 – 1.7 m [36]. Therefore, an aspect ratio of 7:5 (width:height) is adopted to crop the required ROI from the training image dataset to match the animal's general characteristics. The next step is to scale all the cropped images such that all images will have same size. The size of the images in dataset varies from 32×32 to 780×520 pixels. All these images are normalized to a standard size of 64×64 pixels by either up-scaling or down-scaling using conventional bi-cubic interpolation technique. Furthermore, to effectively apply an edge detector to extract the silhouette of the animal, it is required to have at least 10% margin area around the body of the animal.

2) Silhouette and Edge Detection: Most DVCs occurs at night-time due to poor illumination and limited FOV. Moreover, it is difficult to detect deer in night-time as the detection scenario for day and night-time is significantly different. This is due to the fact that the camera used to capture objects in daytime cannot be used for night-time vision.

A thermal or infrared camera is often used to capture objects in the dark as it uses the infrared radiation emitted from the object to create an image. More specifically a camera with a wavelength of 14000nm is used to capture the image of animals in the dark. Therefore, thermal images can be used to train the model for night-vision. The dataset which is used in this work has 2150 infrared thermographic positive images and 11450 negative images. To avoid overfitting the model, it is required to have more images to train the model for night-vision. It is a very challenging task to collect thermographic images of deer with different postures on road and hence it is proposed to train the model with the silhouettes of images captured using regular daytime cameras in addition with the images in the dataset. The intuition is that the silhouette of the animal remains quite similar in both thermal and normal RGB image. Therefore, as a pre-processing step the silhouettes of images in the dataset is obtained using canny edge detection algorithm. These silhouettes are used to train the model. A few silhouettes images of both positive and negative dataset is shown in Fig. 7.

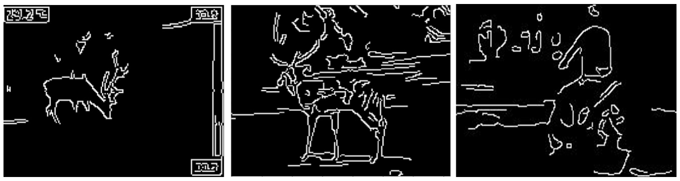


Fig. 7. Sample silhouette images from both positive and negative dataset

C. Convolutional Neural Network (CNN)

A CNN will have three layers viz. convolution layer, pooling layer and the optional fully connected (FC) layers. The convolutional layers are used to learn image-level features from the input images. The dense layer that is at the top of the network will learn very high-level features and combines them to predict and classify an object.

The CNN architecture used in this work is shown in Fig. 8. It consists of three convolutional layers stacked with Rectifying Linear Unit (ReLu) activation function. As shown in Table I, the size of the input image layer is $64 \times 64 \times 3$ and the 1^{st} convolutional layer uses 16 filters with kernel size $3 \times 3 \times 16$. The kernel is slided along the input image both in horizontal and vertical direction at a stride of 1×1 pixels. The filters are used as feature identifiers and it extracts the primitive features in the image such as edges and curves. Zero padding is done on both rows and columns to match the size of the previous layer such that a feature map of size $16 \times 16 \times 64$ is obtained from 1^{st} convolutional layer.

A ReLU activation layer is stacked with the 1^{st} convolutional layer to improve the processing speed of the network [37]. ReLU is a linear activation function and it is defined as

$$y = max(0, x) \tag{1}$$

Whereas x and y are the values of input and output, respectively. It improves the training process by decreasing the vanishing gradient problem [38] which arises due to the use of sigmoid or hyperbolic tangent function during back-propagation.

To improve the robustness of the CNN with respect to translations and noises, the feature maps obtained from the 1^{st} convolutional layer is passed through a max-pooling layer as it is translation invariant. Max pooling layer performs a down-sampling process to reduce the dimensionality of the feature map and hence improves the overall computational efficiency of the network. In this work, the 1^{st} convolutional layer is max pooled using a filter of size 2×2 with a stride of 2 such that we obtain 16 feature maps with a size of 32×32 pixels as shown in Table I.

As shown in Fig. 8 and Table I, the 2^{nd} convolutional layer uses 32 filters with a kernel size of $3 \times 3$and a stride of 1 with necessary zero padding along rows and columns to preserve the size as that of the max pooling layer 1. This layer is also stacked with a ReLU activation layer and its output is max-pooled using a filter of size 2×2 with a stride of 2 such that the size of the feature map is $16 \times 16 \times 32$ as shown in Table I. The first two convolutional layers are used to learn the low-level image features that characterizes a deer. The 3^{rd} convolutional layer is used to extract complex high-level image features. As shown in Table I, it uses 64 filters with a kernel size of 3×3 at a stride of 1. Furthermore, necessary zero padding is done to preserve the size of the feature map as in the previous layer such that the size of the feature map obtained from this layer is $16 \times 16 \times 64$. This layer is also stacked with a ReLU activation layer and its output is max-pooled using a filter of size 2×2 with a stride of 2 such that the size of the feature map is $8 \times 8 \times 64$ as shown in Table I and Fig. 8.

Once the image has passed through all the hidden layers (convolutional and max pooling layers), it is then processed by a fully connected layer. The hidden layers extract all the vital high-level features from the input images to classify it to be an image with deer in it or not. These feature maps were then fed to fully connected layers to classify the class of the object. This layer takes an input volume from the max pooling layer and gives an N-dimensional vector as output. The way this fully connected layer works is that it looks at the output of the previous layer (which should represent the activation maps of high level features) and determines which features most correlate to a particular class. For instance, if an image is classified as deer, it will have high values in the activation maps that represent high level features like antlers or 4 legs, etc. Similarly, if an image is predicted as a background with

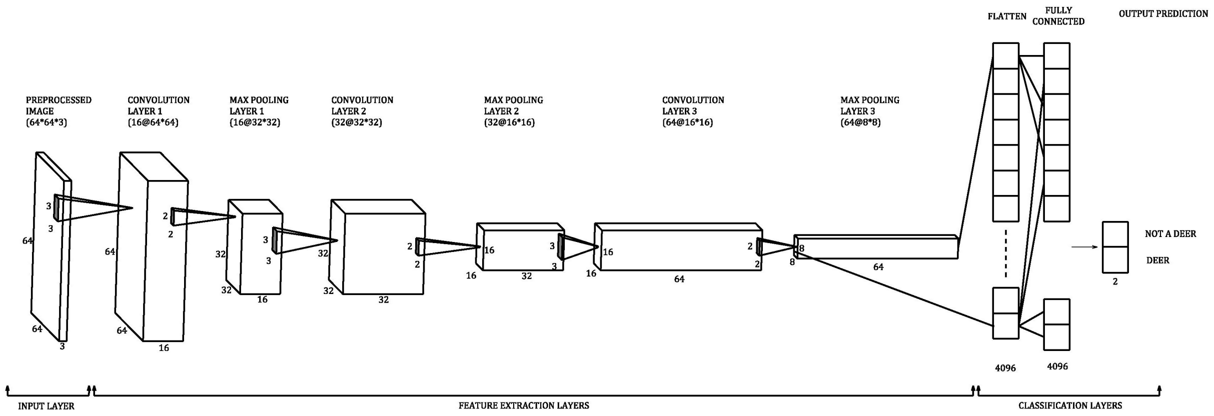

PREPROCESSED IMAGE (64*64*3)
CONVOLUTION LAYER 1 (16@64*64)
MAX POOLING LAYER 1 (16@32*32)
CONVOLUTION LAYER 2 (32@32*32)
MAX POOLING LAYER 2 (32@16*16)
CONVOLUTION LAYER 3 (64@16*16)
MAX POOLING LAYER 3 (64@8*8)
FLATTEN
FULLY CONNECTED
OUTPUT PREDICTION
3
3
64
64
3
64
64
16
2
2
32
32
16
3
3
32
32
32
2
2
16
16
32
3
3
16
16
64
2
2
8
8
64
4096
4096
NOT A DEER
DEER
2
INPUT LAYER
FEATURE EXTRACTION LAYERS
CLASSIFICATION LAYERS

Fig. 8. Proposed CNN architecture for deer detection

TABLE I. Parameters used in the proposed CNN Architecture

Layer Type	Size	k Number of Filters	k Number of Strides	Size of Kernel
Input layer	$64 \times 64 \times 3$			
1st Convolutional layer	$64 \times 64 \times 16$	16	1	3×3
ReLu Layer	$64 \times 64 \times 16$			
Max pooling layer 1	$32 \times 32 \times 16$	1	2	2×2
2nd Convolutional layer	$32 \times 32 \times 32$	32	1	3×3
ReLu Layer	$32 \times 32 \times 32$			
Max pooling layer 2	$16 \times 16 \times 32$	1	2	2×2
3rd Convolutional layer	$16 \times 16 \times 64$	64	1	3×3
ReLu Layer	$16 \times 16 \times 64$			
Max pooling layer 3	$8 \times 8 \times 64$	1	2	2×2
Fully Connected Layer	4096			
Dropout Layer	4096			
Softmax Layer	2			
Classification Layer (Output Layer)	2			

a tree in it, it will have high values in the activation maps that represent high level features like leaves or branches etc. A Fully connected layer looks at what high level features most strongly correlate to a particular class and assign a particular weights to it. In the proposed CNN architecture, the dropout value is experimentally fixed to 0.3. The dropout value is initially fixed to 0.2 and gradually increased up to 0.5 and the accuracy and loss functions are evaluated. It is observed that for the minimum dropout value of 0.2 had very minimal effect on the training and too high value for dropout significantly affected the learning process. The optimal value is fixed as 0.3

Each layer in the CNN architecture will have two kind of parameters viz. weights and biases. The total number of parameters (P) used in CNN architecture is the sum of both weights (W) and biases (B). A detailed summary of parameters used in the proposed CNN architecture is tabulated in Table II.

In this research, the deer (foreground) and the background (Motorcycles, cars and trees) area is classified into two classes using the CNN. As shown in Fig. 8 and Table I, 64 feature maps with 8×8 pixels were obtained after the 3^{rd} convolutional and max pooling layers. The feature maps are flattened into a vector with 4096 elements ($8 \times 8 \times 64$). These elements containing the pixel values are fed into 4096 neurons to form a fully connected layer. The fully connected layer is matrix multiplied with array of weights (4096×2, where 2 is the number of class labels) to produce an output array containing 2 values to predict the output. Learning of weight are done using back propagation method [39]. The initial weights are assigned with random numbers. The feed forward network gives the output value for these weights. For the right class, the probability will be near to 1. The loss between the predicted class and the actual class value is found and the weights are optimized. A softmax function is used to find the probability of each class label [40]. It is given by

$$\sigma(p)_j = \frac{e^{p_j}}{\sum_{k=1}^{K} e^{p_k}} \tag{2}$$

where p_j is the probability of correct class (deer) and p_k is the probability of other classes (background). The softmax activation function is used at the final layer. It provides the probability that the image contains a ‘deer’ or ‘no deer’.

IV. Experimental Results

In this research, a two-class CNN model classifying a deer from its background is developed. The background will have a few classes of objects such as motorcycles, cars and trees which are frequently encountered on roadways. To achieve this, we initially developed a simple two-class model which can differentiate deer with motorcycles. This model is then

TABLE II. A DETAILED SUMMARY OF THE NUMBER OF PARAMETERS USED IN VARIOUS LAYERS OF THE PROPOSED CNN MODEL

Layer Name	Weights	Bias	Parameters
Input layer	0	0	0
1st convolutional layer	432	16	448
Max pool layer 1	0	0	0
2nd convolutional layer	864	32	896
Max pool layer 2	0	0	0
3rd convoolutional layer	1728	64	1792
Max pool layer 3	0	0	0
Fully connected layer	16777216	4096	16781312
Output layer	0	0	0
Total	16780240	4208	16784448

extended to a multi-class model having four object classes viz. deer, motorcycles, cars and trees. Finally, a two-class model containing two classes, namely, 'deer' and 'not a deer' is developed with 'not a deer' class having a combination of background with motorcycles, cars and trees. Moreover, the CNN model is trained using three sets of input images viz. RGB color images, thermal images and silhouettes of the image to evaluate its performance in both day and night vision.

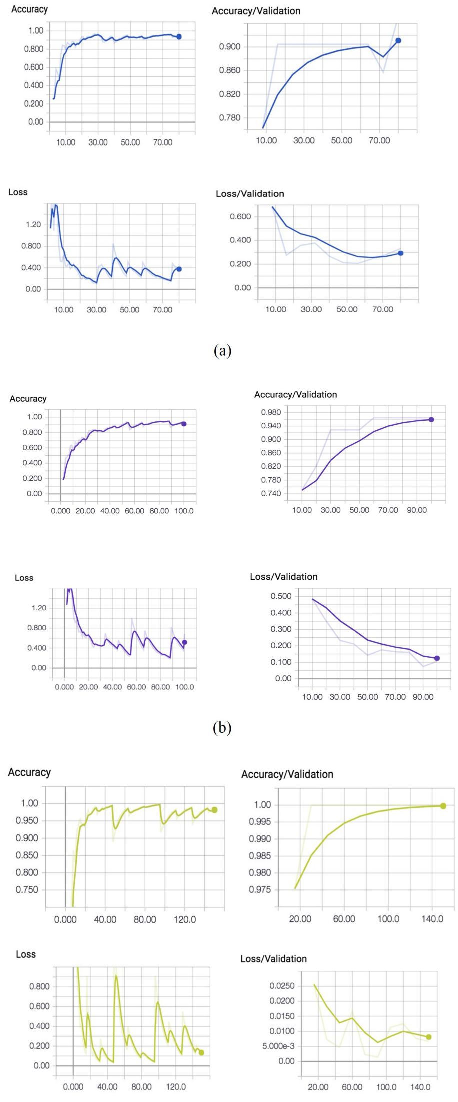
A. Experimental Setup

The proposed CNN architecture is implemented on top of Tensorflow, an open source deep learning library created by Google. The training of CNN model is carried out using a desktop computer with Intel core i5-2400@2.7GHz processor with 8 GB RAM. The proposed CNN model is trained for both day and night-time vision using the self-constructed dataset as mentioned in Section 3.1. The models are compared based on a few metrics including training and validation curves for loss and accuracy, test set accuracy on different classes as well as classification time.

B. Proposed Multi-Class CNN Model

It is required to consider multiple classes of objects that are frequently encountered on roadways to accurately detect the presence of deer on road. Therefore, a multi-class CNN model is developed to improve the effectiveness of the proposed system. To achieve this, the two class CNN model proposed to differentiate deer and motorcycles is extended to a multi-class model which differentiates deer from a background. The background class will have four sub-classes of objects namely motorcycles, cars, pavements and trees. Thus, the final CNN model will be a two-class model having 'deer' and 'not a deer' classes with 'not a deer' having any of the four sub-classes of the background.

1) Accuracy and Loss curves: The accuracy and loss curve of the multi-class CNN model trained only with RGB color images for 80 iterations is shown in Fig. 9 (a). This model is trained with 800 images from each class with a batch size of 200. The average accuracy of this model is approximately 90% which is slightly lesser than the two-class model. Moreover, the validation accuracy is significantly lesser which can be observed in the loss curve as well. A loss of approximately 20-25% is seen during the validation process. The relatively lower accuracy during validation process is attributed to the fact that increasing the number of classes makes it harder to differentiate the classes. Furthermore, this model is trained only with color images and hence will work only on daytime



(a)

(b)

(c)

Fig. 9. Training and validation curves for multi-class model trained with (a) color images (b) Thermal images (c) Silhouette images

images. For night-time vision, the multi-class model is trained with approximately 400 thermal images from each class. Fig. 9 (b) depicts the training and validation curves for accuracy and loss of this model for 80 iterations. It is observed that the peak accuracy of this model is less than 90% during both training and testing phase. Moreover, the accuracy is much lesser during the initial iterations and only after 50 iteration the convergence of this model is satisfactory. Hence, multi-class thermal model is not satisfactory for on-road deer detection during night-time. On the contrary, the multi-class CNN model trained with the silhouettes of the image achieves a high accuracy compared with the other two models. The CNN is trained with the silhouettes of images obtained by canny edge detection technique. The graphical representations of the accuracy and loss is depicted in Fig. 9(c). The CNN is trained with approximately 1000 silhouette images of four classes of objects. This CNN model trained with silhouettes fared much better when compared to the multi-class thermal classifier model. An average accuracy of 98% was obtained while trying to classify deer using this model. Moreover, the convergence speed of this model is very high compared with the other two approaches. The high accuracy of this model is attributed to the fact that silhouettes serve as a cue to learn significant features which characterizes an object class. Furthermore, this model perform much better both in day and night-time conditions due to the fact that silhouettes of deer is similar both in color and thermal images.

2) Detection of deer using sliding window approach: The next step after classifying an image is to localize the spatial region that contains a deer in it. This is achieved by sliding a fixed size window from top-left corner of an image to the bottom-right position of an image. Furthermore, to detect the presence of deer in different scales an image pyramid is created by up-scaling and down-scaling the given image by a factor of 2. The sliding window approach for deer localization is depicted in Fig. 10.

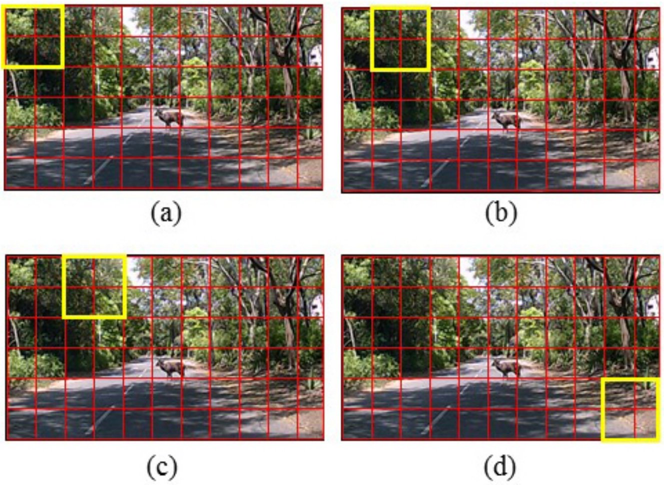
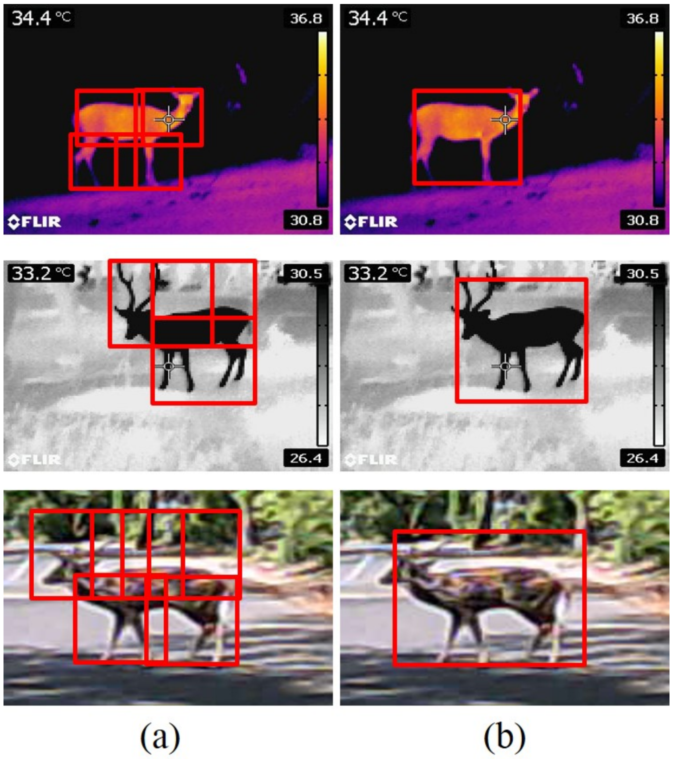

(a)
(b)
(c)
(d)

Fig. 10. Sliding window approach for deer localization (a) first sliding window (b) second sliding window (c) Third sliding window (d) last sliding window

The yellow box in Fig. 10 depicts the sliding window that is slided over the entire image. The ROI is extracted from each stop of the sliding window. The extracted ROI is fed to a pre-trained CNN model and if its classification probability is higher than the threshold for the class 'deer', then the ROI is labeled as 'deer'. This process is repeated for all the spatial location of the image. Finally, all ROI with label as 'deer' are grouped and the one with the highest probability is marked as 'deer' using non-maxima suppression. Fig. 11 shows a few examples of this approach to detect deer.



(a)
(b)

Fig. 11. A few examples for localization of deer using sliding window approach (a) without non-maxima suppression (b) with non-maxima suppression

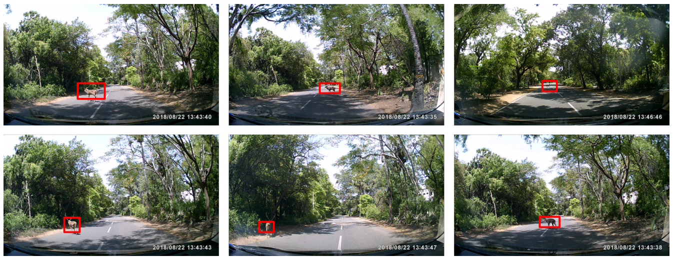


Fig. 12. Experimental results performed on images captured with a dashboard camera

Fig. 12 and Fig. 13 shows a few experimental results of deer detection performed on images captured with dashboard camera and thermal images respectively. The dashboard camera is installed in a car and the image frames extracted from video are evaluated using the proposed CNN classifier. It is observed that the pre-trained CNN model effectively detects the presence of deer and highlights it within a bounding box shown in red color. Furthermore, it is evident from Fig. 13 that the proposed model is capable of detecting deer in night-time as well.

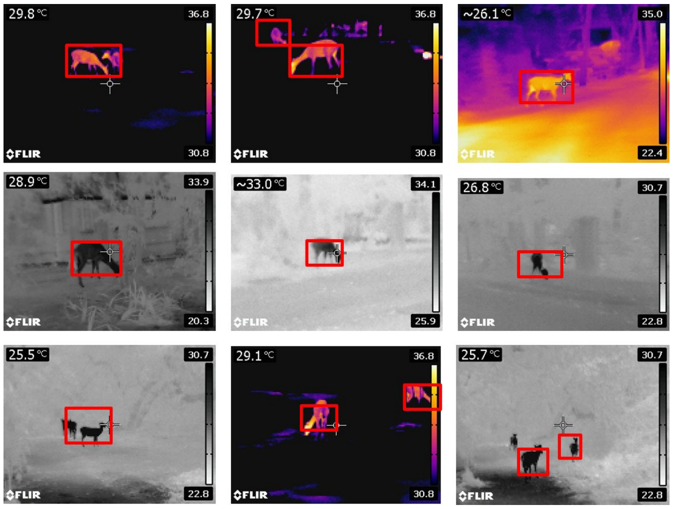


Fig. 13. Night-time deer detection performed on thermal images

C. Evaluation of the Proposed Model

The proposed CNN based Deer detection algorithm consists of a five layer CNN classifier to detect the presence of deer. The performance metrics of the proposed CNN model is evaluated in this section.The proposed CNN model is evaluated based on a few metrics such as classification loss, localization loss, and total loss.

a) Classification loss: Classification loss gives the performance measure of an object detection model where the probability distribution of the output varies between [0,1]. It is the loss associated with the classification of detected objects into various classes. It is also known as cross-entropy loss and it represents the price paid for inaccuracy of predictions in classification problems. The bounds for the classification loss are defined by Bayes' Theorem. This loss increases as the predicted probability diverges from the actual label. Cross-entropy loss penalizes heavily the predictions that are confident but wrong. The classification loss for the proposed method is shown in Fig. 14. The classification loss for the proposed method converges at 85K steps.

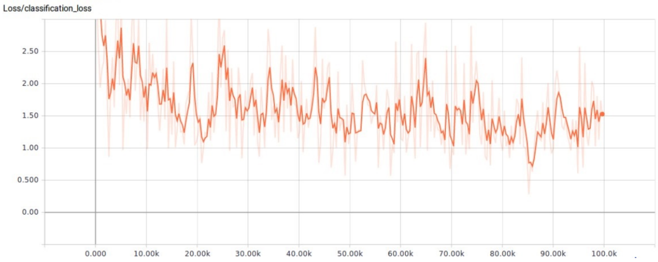


Fig. 14. Classification loss curve of the proposed model

b) Localization loss: Localization loss is the measure of mismatch between the ground truth bounding box and the predicted bounding box. The localization loss is obtained only from positive match predictions. The negative matches are ignored in calculating the localization loss. A lesser localization loss infer that the predicted bounding box is closer to the ground truth bounding box. The localization loss for the proposed method is shown in Fig. 15. It is seen that the proposed method localizes the object of interest with more accuracy at the end of training.

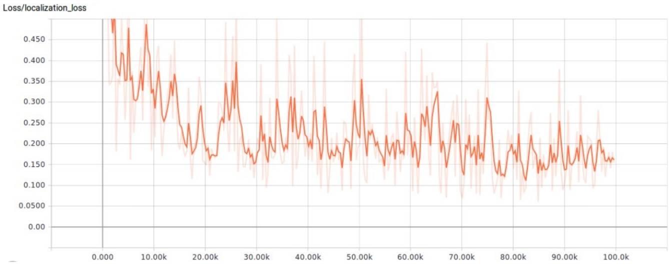


Fig. 15. Localization loss curve of the proposed method

c) Total Loss: Total loss is a step-wise summation of both classification and localization loss. This parameter provides an overall prediction loss for the chosen model. The total loss for the proposed model is shown in Fig. 16. It is observed that the total loss is less than 10% and therefore can be efficiently used for the detection of deer on roadways.

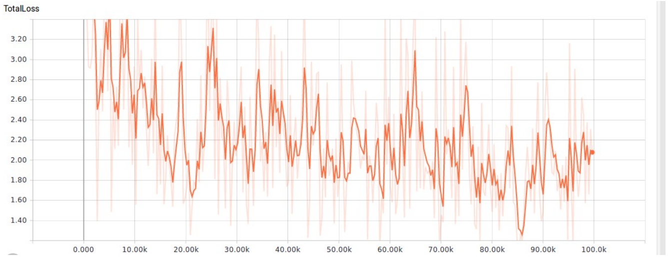


Fig. 16. Total loss curve of the proposed method

Comparison with state-of-the-art classifiers: To evaluate the classification performance of the proposed CNN model with other state-of-the-art approaches, a few parameters such as positive predictive value (PPV), True positive rates (TPR), Accuracy (ACC) and F_Score are used. The detailed comparison of the above parameters for the proposed method with other state-of-the-art approaches such as HoG-AdaBoost classifier, LBP-AdaBoost classifier, Haar-AdaBoost classifier and HoG-SVM classifier is presented in Table III.

Based on the TP, TN, FP and FN shown in Table III , the following criteria are used to assess the accuracy of the classifier model

$$\text{Positive Predicitive Value (PPV)} = \frac{\#TP}{\#TP + \#FP} \tag{3}$$

$$\text{True Positive Rate (TPR)} = \frac{\#TP}{\#TP + \#FN} \tag{4}$$

$$\text{Accuracy}=\frac{\#TP + \#TN}{\#TP + \#TN + \#FP + \#FN} \tag{5}$$

$$F\_Score = 2 \times \frac{PPV \times TPR}{PPV + TPR} \tag{6}$$

TABLE III. CONFUSION MATRIX FOR RECOGNITION ACCURACIES FOR VARIOUS STATE-OF-THE-ART APPROACHES (A) HOG-ADABOOST (B) LBP-ADABOOST (C) HAAR-ADABOOST (D) HOG-SVM (E) PROPOSED CNN MODEL

(a)			
HoG-AdaBoost		**Recognized**	
		Deer	**Background**
Actual	**Deer**	0.95	0.05
	Background	0.0186	0.9813
(b)			
LBP-AdaBoost		**Recognized**	
		Deer	**Background**
Actual	**Deer**	0.9709	0.0290
	Background	0.0136	0.9869
(c)			
Haar-AdaBoost		**Recognized**	
		Deer	**Background**
Actual	**Deer**	0.9627	0.0372
	Background	0.0172	0.9827
(d)			
HoG-SVM		**Recognized**	
		Deer	**Background**
Actual	**Deer**	0.9445	0.0554
	Background	0.0218	0.9781
(e)			
Proposed Model		**Recognized**	
		Deer	**Background**
Actual	**Deer**	0.9881	0.0118
	Background	0.0095	0.9904

Where, #TP, #TN, #FP, and #FN are the mean of the number of TP, TN, FP and FN, respectively. Based on these, the accuracies of the classifier model is evaluated and is presented in Table IV and Fig. 17.

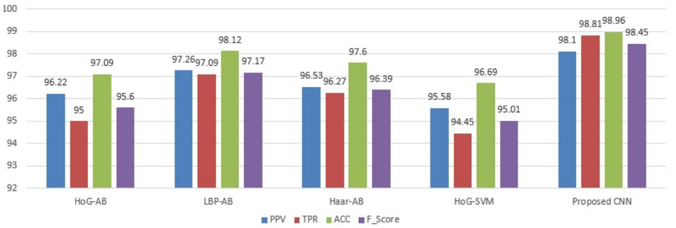


Fig. 17. Comparison of accuracy scores for various state-of-the-art approaches (unit %)

From Table IV and Fig. 17 it is observed that the accuracy score of the proposed CNN model is better than other state-of-the-art approaches.

Comparison on Average Detection Time: The time required by the model to classify an image as deer or not is evaluated by its average detection time. A comparison is made with other state-of-the-art classifiers such as LBP-AdaBoost, HoG-SVM, Haar-AdaBoost and HoG-AdaBoost classifier with the proposed CNN classifier. Fig. 18 shows the average detection time for various state-of-the-art classifiers. It is observed that the HoG with SVM classifier consumes more time (150ms) compared with other approaches. This is due to the fact that the time required to extract the HoG features for a larger frame is time-consuming compared with Haar and LBP features. However, by using an AdaBoost classifier the average detection time has significantly reduced. However, it is seen from Fig. 18 that the average detection time of CNN classifier is much lesser (27.1ms). It is due to the fact that once the CNN is trained and its weights are set, the classification task is very fast. The processing time for the CNN classifier is tested on 1000 images. It is observed that the average detection time is 27.1 ms.

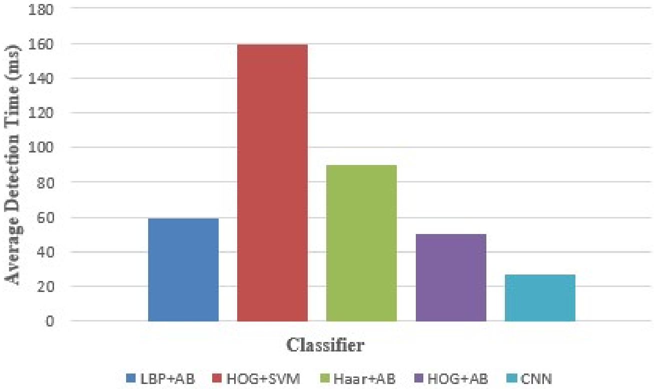


Fig. 18. Average detection time of various state-of-the-art classifiers

Comparison with pre-trained CNN models: In this research work, we have compared the performance of the proposed method with three most common and successful CNN architectures namely AlexNet [41] , VGG-16 [42] and ResNet-50 [43]. For the experiments, a simplified version of AlexNet which has eight layers with three convolutional layers is used. followed by two fully connected layers. The state-of-the-art VGG-16 is much denser with 13 convolutional layer and three fully connected layers. Moreover, ResNet-50 which is much deeper than VGG_16 is also used for the comparison.

The performance metrics used in this research are Accuracy, F_Score, and the inference time. The accuracy and F_score of the models are calculated based on the number of True Positive (TP), True Negative (TN), False Positive (FP) and False negative (FN) cases reported by a model. The state-of-the-art models are trained with the proposed dataset from the scratch and are evaluated against the proposed model.

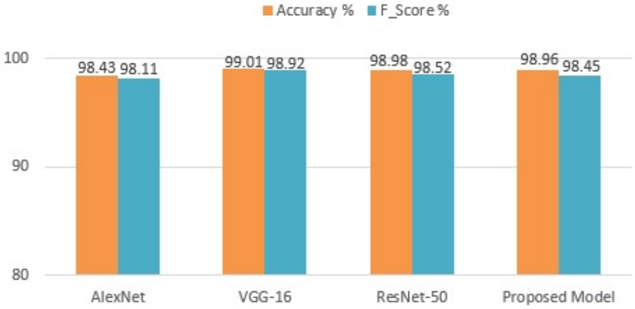


Fig. 19. Comparison of accuracy metrics of state-of-the-art pre-trained models with the proposed model

Having trained all the models from the scratch, it is interesting to note that all three pre-trained model show similar accuracy and F_score as reported in Table V.

Furthermore, it is observed that the accuracy metrics of VGG-16 is slightly better than other models. Moreover, the accuracy and F_score of the proposed model is at par with

TABLE IV. A DETAILED SUMMARY OF COMPARISON OF ACCURACY SCORES FOR VARIOUS STATE-OF-THE-ART CLASSIFIER APPROACHES (UNIT %).

Classifier	PPV	TPR	Accuracy	F_Score
HoG+AB	96.22	95	97.09	95.60
Lbp+AB	97.26	97.09	98.12	97.17
Haar+AB	96.53	96.27	97.60	96.39
HoG+SVM	95.58	94.45	96.69	95.01
Proposed CNN	98.10	98.81	98.96	98.45

TABLE V. PERFORMANCE COMPARISON OF THE PROPOSED MODEL WITH STATE-OF-THE-ART PRE-TRAINED MODELS

Model	Trainable Layers	Accuracy%	F_Score	Inference Time (ms)
AlexNet	8	98.43	98.11	68
VGG-16	16	99.01	98.92	156
ResNet-50	50	98.98	98.52	62
Proposed Model	4	98.96	98.45	57

ResNet-50. A comparison of accuracy and F_score of the proposed model with state-of-the-art pre-trained model is shown in Fig. 19.

To compare the inference time, the training of aforementioned models are performed on Nvidia Geforce GTX 750Ti equipped with an Intel Core i5 4440S and 12GB RAM. It is observed from Table V that VGG-16 despite having the best accuracy metrics, its computational time is much higher compared to other models. On contrary, the much denser ResNet with 50 layers is computationally efficient with its inference time similar to AlexNet's inference time. Moreover, it is observed that the inference time for the proposed model is low with 57 ms. This corroborates that the proposed model can be implemented without a GPU and therefore it can be used for on-road deer detection efficiently.

V. CONCLUSION

To mitigate the severity of DVC, a CNN based methodology to detect deer on roadways is presented in this paper. A multi-class CNN classifier is trained to classify an image based on the presence of deer. The proposed model can effectively differentiate a deer from its background. The background has four sub-classes of objects that are frequently encountered in roadways such as motorcycles, cars, pavements and trees. A large self-constructed positive dataset with images of deer in different poses and time is created. Moreover, to make the model robust, a negative dataset with objects other than deer is also created. The proposed CNN model is trained for both daytime and night-time vision using RGB color images and thermal images respectively. However, owing to the limitation in capturing night-time images of deer using a thermal camera, we propose a method to train the CNN using the silhouettes of the images obtained using an edge detection technique. A detailed performance evaluation is carried out on the three models and it is observed that the CNN model trained using the silhouettes of the images have better classification accuracy. Furthermore, the spatial region having deer in it is localized using sliding window approach. The aforementioned technique is evaluated on a variety of test images and the results benchmarks the effectiveness of the proposed technique.

REFERENCES

[1] K. C. Dey, A. Mishra, and M. Chowdhury, "Potential of Intelligent Transportation Systems in Mitigating Adverse Weather Impacts on Road Mobility: A Review," *IEEE Transactions on Intelligent Transportation Systems*, vol. 16, no. 3, pp. 1107–1119, June 2015.

[2] K. Bengler, K. Dietmayer, B. Farber, M. Maurer, C. Stiller, and H. Winner, "Three Decades of Driver Assistance Systems: Review and Future Perspectives," *IEEE Intelligent Transportation Systems Magazine*, vol. 6, no. 4, pp. 6–22, 2014.

[3] R. Benenson, M. Omran, J. Hosang, and undefined B. Schiele, "Ten years of pedestrian detection, what have we learned?" in *European conference on computer vision*, 2014, pp. 613–627.

[4] P. Hurney, P. Waldron, F. Morgan, E. Jones, and M. Glavin, "Review of pedestrian detection techniques in automotive far-infrared video," *IET Intelligent Transport Systems*, vol. 9, no. 8, pp. 824–832, 2015.

[5] J. M. Conn, J. L. Annest, and A. Dellinger, "Nonfatal motor-vehicle animal crash-related injuries, United States, 2001-2002,," *Journal of Safety Research*, vol. 35, no. 5, pp. 571–574, 2004.

[6] W. E. Hughes, A. R. Saremi, and J. F. Paniati, "Vehicle-Animal Crashes: An Increasing Safety Problem," *ITE Journal*, vol. 66, no. 8, pp. 24–28, 1996.

[7] S. R. Meister, M. M. Hing, W. G. M. Vanlaar, and R. D. Robertson, *Road Safety Monitor 2014: Driver Behaviour and Wildlife on the Road in Canada*. Ottawa, Ontario: Traffic Injury Research Foundation, 2016.

[8] N. Putzu, D. Bonetto, V. Civallero, S. Fenoglio, P. G. Meneguz, N. Preacco, and P. Tizzani, "Temporal patterns of ungulate-vehicle collisions in a subalpine Italian region," *Italian Journal of Zoology*, vol. 81, no. 3, pp. 463–470, 2014.

[9] J. Mrtka and M. Borkovcová, "Estimated mortality of mammals and the costs associated with animal-vehicle collisions on the roads in the Czech Republic," *Transportation research part D: transport and environment*, vol. 18, pp. 51–54, 2013.

[10] "Ministry of Home Affairs. Accidental Deaths & Suicides in India 2006, Nat. Crime Records Bureau," New Delhi, India, 2007.

[11] "Ministry of Home Affairs. Accidental Deaths & Suicides in India 2012, Nat. Crime Records Bureau," New Delhi, India, 2013.

[12] M. R. Conover, W. C. Pitt, K. K. Kessler, T. J. DuBow, and W. A. Sanborn, ""Review of human injuries, illnesses, and economic losses caused by wildlife in the United States"," *Wildlife Society Bulletin*, vol. 23, no. 3, pp. 407–414, 1973.

[13] H. H. James, D. C. Paul, C. Gwen, and F. W. Allan, "Methods to Reduce Traffic Crashes Involving Deer: What Works and What Does Not," *Traffic Injury Prevention*, vol. 5, no. 2, pp. 122–131, 2004.

[14] H. Torsten, M. Jörg, H. Leonhard, M. Lisa, and M. Atle, "Temporal patterns of deer–vehicle collisions consistent with deer activity pattern and density increase but not general accident risk," *Accident Analysis & Prevention*, vol. 81, pp. 143–152, 2015.

[15] A. F. Williams and J. K. Wells, "Characteristics of vehicle-animal crashes in which vehicle occupants are killed," *Traffic Injury Prevention*, vol. 6, no. 1, pp. 56–59, 2005, PMID: 15823876.

[16] M. A. Sharafsaleh, *Evaluation of an animal warning system effectiveness phase two-final report,*. Berkeley, CA, USA: Tech, 2012.

[17] K. Knapp, “Deer-vehicle crash countermeasure toolbox: A decision and choice resource,,” *Midwest Regional*, 2004.

[18] D. Zhou, “Real-time Animal Detection System for Intelligent Vehicles,” 2014.

[19] A. Mammeri and D. A. Boukerche, “”Animal-Vehicle Collision Mitigation System for Automated Vehicles,” ,” *IEEE Transactions on Systems, Man, and Cybernetics: Systems*, vol. 46, no. 9, pp. 1287–1299, 2016.

[20] P. Dollar, C. Wojek, and B. P. Perona, “Pedestrian Detection: An Evaluation of the State of the Art,” *IEEE Transactions on Pattern Analysis and Machine Intelligence*, vol. 34, no. 4, pp. 743–761, 2012.

[21] F. Li, R. Zhang, and F. You, “Fast pedestrian detection and dynamic tracking for intelligent vehicles within V2V cooperative environment,” *IET Image Processing*, vol. 11, no. 10, pp. 833–840, 2017.

[22] A. Satpathy and X. H. Eng, “LBP-Based Edge-Texture Features for Object Recognition,” *IEEE Transactions on Image Processing*, vol. 23, no. 5, pp. 1953–1964, 2014.

[23] L. Zhang, R. Chu, S. Xiang, S. Liao, and undefined S. Z. Li, “Face detection based on multi-block LBP representation,” *Advances in Biometrics*, pp. 11–18, 2007.

[24] Y. Zhao, Y. Zhang, R. Cheng, and D. G. Li, “An Enhanced Histogram of Oriented Gradients for Pedestrian Detection,” *IEEE Intelligent Transportation Systems Magazine*, vol. 7, no. 3, pp. 29–38, 2015.

[25] R. Girshick, J. Donahue, and T. J. Malik, “Region-Based Convolutional Networks for Accurate Object Detection and Segmentation,” *IEEE Transactions on Pattern Analysis and Machine Intelligence*, vol. 38, no. 1, pp. 142–158, 2016.

[26] A.Rabinovich, W. L. C.Szegedy, P. S. Y.Jia, D. A. S.Reed, and V. V. D.Erhan, “Going deeper with convolutions,” in *CVPR*, 2015.

[27] W. Ouyang and X. Wang, “Joint Deep Learning for Pedestrian Detection,” in *Computer Vision and Pattern Recognition*, 2013.

[28] T. Burghardt and J. Calic, “Real-time face detection and tracking of animals,,” in *Proc. 8th Seminar Neural*, 9 2006, pp. 27–32.

[29] S. Paisitkriangkrai and C. J. Zhang, “Performance evaluation of local features in human classification and detection,” *IET Computer Vision*, vol. 2, pp. 236–246, 2008.

[30] D. Ramanan, D. A. Forsyth, and undefined K. Barnard, “Building models of animals from video,” *IEEE Trans. Pattern Anal. Mach. Intell*, vol. 28, no. 8, pp. 1319–1334, 2006.

[31] T. Burghardt, B. Thomas, P. J. Barham, and J. Calic, “Automated visual recognition of individual African penguins,,” in *Proc. 5th Int. Penguin Conf*, Ushuaia, Argentina, 9 2004.

[32] Z. Debao, W. Jingzhou, and undefined W. Shufang, “Countour based HOG deer detection in thermal images for traffic safety,” in *Proc. Int. Conf. Image Process. Comput. Vis. Pattern Recognit., Las Vegas*, NV, USA, 7 2012, pp. 1–6.

[33] W. Zhang, J. Sun, and undefined X. Tang, “From tiger to panda: Animal head detection,” *IEEE Trans. Image Process*, vol. 20, no. 6, pp. 1696–1708, 2011.

[34] M. Zeppelzauer, “Automated detection of elephants in wildlife video,” *EURASIP J. Image Video Process*, vol. 46, no. 1, pp. 1–44, 2013.

[35] P. Khorrami, J. Wang, and undefined T. Huang, “Multiple animal species detection using robust principal component analysis and large displacement optical flow,,” in *Proc. Workshop Vis. Observation Anal. Animal Insect Behav*, VAIB), Tsukuba, Japan, 2012.

[36] R. Tharmalingam, S. Kalyanasundaram, Q. Qamar, and K. Riddhika, “Group size, sex and age composition of chital (Axis axis) and sambar (Rusa unicolor) in a deciduous habitat of Western Ghats,” *Mammalian Biology - Zeitschrift für Säugetierkunde*, vol. 77, no. 1, pp. 53–59, 2012.

[37] V. Nair and G. E. Hinton, “Rectified linear units improve restricted boltzmann machines,” in *ICML’10: Proceedings of the 27th International Conference on Machine Learning*, Haifa, Israel, 6 2010, pp. 21–24.

[38] X. Glorot, A. Bordes, and Y. Bengio, “Deep sparse rectifier neural networks,” in *14th International Conference on Artificial Intelligence and Statistics, 11–13 April 2011, USA*, FL, USA, 4 2011, pp. 315–323.

[39] G. E. Hinton and R. R. Salakhutdinov, “Reducing the Dimensionality of Data with Neural Network,” *Science*, vol. 313, no. 5786, pp. 504–507, 2006.

[40] W. Liu, Y. Wen, Z. Yu, and M. Yang, “Large-Margin Softmax Loss for Convolutional Neural Networks,” in *ICML’16: Proceedings of the International Conference on Machine Learning*, vol. 21, 2016, pp. 765–789.

[41] A. Krizhevsky, I. Sutskever, and G. E. Hinton, “ImageNet Classification with Deep Convolutional Neural Networks,” in *Proceedings of the 25th International Conference on Neural Information Processing Systems - Volume 1*. USA: Curran Associates Inc., 2012, pp. 1097–1105.

[42] K. Simonyan and A. Zisserman, “Very Deep Convolutional Networks for Large-Scale Image Recognition,” *CoRR*, vol. abs/1409.1556, 2014.

[43] K. He, X. Zhang, S. Ren, and J. Sun, “Deep Residual Learning for Image Recognition,” in *2016 IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*, June 2016, pp. 770–778.