Deep Learning Methods for monitoring, detecting and measuring Deer Movements for Wildlife Conservation

Neelam Rawat, J.S. Sodhi, Rajesh K. Tyagi

Abstract: Wildlife Researchers examine and dig video corpus for behavioral studies of free-ranging animals, which included monitoring, analyzing, classifying & detecting, managing, counting etc. Unfortunately, automated visual implementation for challenging real-time scenarios of wildlife is not an easy task especially for classification and recognition of wildlife-animals and estimate the sizes of wildlife populations. The aim of this paper is to bring state-of-the-art results from raw sensor data for learning features advancing automatic implementation and interpreting of animal movements from different perspectives. Also, turnout with an objectness score from object proposals generated by Region Proposal Network (RPN). The imagery data are captured from the motion sensor cameras and then through RCNN, Fast RCNN and Faster RCNN, it automatically are segmented and recognized the object with its objectness score. ConvNet automatically process these images and correctly recognizing the object. Experimentation results demonstrated prominent deer images with 96% accuracy with identifying three basic activities sleeping, grazing and resting. In addition, a measured implementation has been shown among CNN, RCNN, Fast RCNN and Faster RCNN.

Keywords: Wildlife, Ecosystem, Ecology, Deep Learning, Convolutional Neural Network, Region Proposal Network, motion sensor cameras

I. INTRODUCTION

Understanding wildlife conservation is the practice of protecting animal species and balancing the ecosystem so that to provide the sustenance in wildlife preservation and management. Population of living species becoming extinct due to climate change, unregulated hunting and poaching, over-exploitation, deforestation etc. Despite these issues, need of wildlife conservation in India is now questioned. Apparently, several wild animal species, forest brimming towards destruction and disappearance. To have a finer knowledge of wildlife ecosystem, more convincing and virtuous implementation is required for wildlife species [27]. For this, an accurate method for auto-implementation of animal identification, detection and monitoring from camera-trap images should be contrived [28][29]. Effective surveillance is crucial for every detection and successful monitoring for terrestrial animals and their habitats.

From camera-trap images [1], processing of extensive amount of images and videos are very expensive and time consuming which may further be a major obstacle for research scientists and ecologists. In this paper, a legal framework is proposed to frame an auto-implementation animal detection and recognition in the wild. In wildlife monitoring, the standard data used is population-ratio (Figure 2). Statistically, small samples are not reliable, but may be the animals having same habitat and living, in that case, a small sample of animals can represent the population. Monitoring the frequency relying on individual animals in the detection and identification process. Estimating the density of animals without use of individual recognition may also speed-up the underlying process [5][6].

In this paper, a legal framework is proposed to frame an auto-implementation animal detection and recognition in the wild. In wildlife monitoring, the standard data used is population-ratio. Statistically, small samples are not reliable, but may be the animals having same habitat and living, in that case, a small sample of animals can represent the population. Monitoring the frequency relying on individual animals in the detection and identification process [2][3]. Estimating the density (or counting) of animals without use of individual recognition may also speed-up the underlying process [10][26].

From recent years, Deep Learning (DL) techniques proves to be boom in the area of computer vision [17]. The objective of this paper is to do measured implementation on DL methodologies and through these, monitoring and detecting the deer movements and activities. Although, it may frame for any wildlife species, but this multi-task learning method make a clear picture of the frame that able to identify effectively and transit that in model [4][16].

Figure 1: Recognizing Deer with Objectness Scores
Fast RCNN (Fast Region-based Convolutional Neural Network) extracts independent features from each region that improves the performance compare to RCNN and simple CNN because through Fast RCNN computation is now only on an image as a whole [18]. This time, no need to feed thousands of regions to CNN every time but operation is done only once per image to generate feature. (Figure 4). In addition, Region of Interest (RoI) pooling layer will defined directly the shape of each region with height and width specifications that makes RoI layer to extract features of same shapes from RoI of different shapes [19].

Figure 4: Fast RCNN Model with RoI Pooling and Selective Search algorithm for classification

Fast RCNN model achieves excellent object detection accuracy compare to the previous defined models. However, the computational and training powers in terms of space and time is very expensive and the process of detecting objects is quite slow [13].

Faster RCNN (Faster Region-based Convolutional Neural Network) defines the precise detection of objects by replacing selective search to region proposal network (RPN). Through RPN, number of proposed generated regions will be reduced and ensure the object detention process in precise manner [20][21]. Faster RCNN generates high-quality regions from selective search algorithms to RPN including the detection of boundary boxes and category with binary category prediction and bounding box for anchor box in RPN (for predicting whether there is an object or not with defined bounding box for objects). (Figure 5). Therefore, faster RCNN has two main modules: deep fast RCNN for the use of detected regions and CNN for region, which defines a unified model for object detection and identification process [14].

Figure 3: RCNN with defined images and selective search algorithm for classification

RCNN models effectively extract target images from pre-trained CNNs but the computational speed becomes slow as thousands of regions can be formed from a single image which require thousands of computation perform to suitable for real-time applications frequently based on number of regions formed [11][12].

Figure 2: Population Ratio of Deer with Confidence level of detection accordingly.

A. CNN, RCNN, FAST RCNN AND FASTER RCNN

CNN (Convolutional Neural Network) is a class of feed forward artificial neural networks which is designed as a multi-layer perceptron (deep learning algorithm) to identify different objects, recognize each object individually with minimal processing [7][8]. CNN applications are image recognition, text analytics, for sound and graph data etc. CNN not able to perceive images, rather consider images as volumes with its width and height measured by its pixels. For each pixel, some numbers define intensity of R, G, and B and that identified number will be a component among three. These numbers are initially raw features, which are feed in CNN to find the significance of those numbers as objects. This filtering process continues until it identifies the patterns in the pixels [5][6]. The major shortcomings of using CNN are if features are homogeneous then no issues in processing and identifying images but will be slow for different scales of heterogeneous features and ineffective also. CNN is cost effective if have had a large dataset with good GPU because the computational cost of CNN is quite high as works well with high dimensional data.

RCNN (regions with CNN features) or Region-based Convolutional Neural Network that apply to deep models for object recognition [9][10]. RCNN defines the region from any image by using a bounding box so that the focus on a single region at a time to avoid interference. In RCNN, region is defined with selective search algorithm with continual of resizing it so that of equal region sizes before feeding to CNN for classification process (Figure 3). Selective Search selects multiple regions
Further, RoI pooling layer replaced by RoI alignment layer to predict pixel-level positions of objects that is not done in Faster RCNN. So we do have Mask RCNN which modifies the faster RCNN model by using RoI alignment layer for spatial information of an object [15][22]. This pixel-level semantic detection could be a challenge and limitation for faster RCNN.

II. MATERIAL AND METHODS

CNN (Convolutional Neural Network), the simplest deep learning methods for widely used in detecting images. CNN considers as a part of artificial neural networks for identifying different objects and activities with recognizing each object separately of minimal computational processing. CNN is not bound only for image processing, but its efficacy proves to work on medical diagnosis, business pattern detections and so on. Let’s consider firstly the inner working of CNN (Figure 6) as:

Figure 5: Faster RCNN Model with Region Proposal Network (RPN) for classification

Step 1: Consider an image as an input (Figure 7. below).

Step 2: Various regions are formed by splitting the input image (Figure 8. Below).

Step 3: We marked individual image from each region

Step 4: Propel all these regions (images) to CNN and arrange it into various classes.

Step 5: Finally, we detect the object and merge all the regions to get the original image (Figure 9. Below).

Figure 6: CNN Framework

Worryment is the objects in images may have several aspect ratios and spatial locations [23][24]. Such as, in particular, cases, the object might overlay most of the image, but in others object might have a very small percentage of image. In given Figure 6. on left hand side of the image, object covering most of the images compare to other side of the same. In addition to, shape and size of objects may also vary. With the consideration of all these factors, we would require voluminous regions that may result with excessive computational time [25].
Therefore, we need to scale down the number of regions and use region-based CNN that screened the regions in particular.

To avoid such issues, we need to focus on single regions at a time. In RCNN (Region-based Convolutional Neural Network), CNN is coerced to concentrate on a particular region to recede the interferences. RCNN uses Selective Search Algorithms i.e., an object proposal algorithm that uses the indications like textures, intensity, color etc. to predict the possible locations of objects. These generated boxes then go as an input to CNN classifier with the following important aspects:

- Run the algorithm to find all possible objects
- These regions further get feed to CNN so that particular class get detected by SVM
- Optimize the regions for identifying the patterns in the images.

Following are the steps to describe how selective search (SS) for RCNN works:

Step 1: Initially, consider an image as an input.
Step 2: To have multiple regions, segment the images.
Step 3: The technique commingles related regions to form a particular region based on texture, color, size and shape similarity.
Step 4: At last, final location of object is being produced from these regions (i.e., Region of Interest)

Through Selective Search (SS) algorithm and detection network get uncoupled, through, which if error occurred in SS network then it, may affect the detection network directly. So it is preferred that SS and detection network correlate with each other. However, training with RCNN is very expensive and slow as well because if we consider N images and 1000 features for extraction from them by using CNN, then it needs N*1000 CNN features which may make RCNN very slow. As around 50-60 seconds of time will take to have a prediction of individual images that make it a cumbersome model and practically not possible to host.

We can have all ROI by executing once for each image through Fast RCNN (Fast Region-based Convolutional Neural Network) because Fast RCNN uses RPN at feature map level instead pixel level. In Fast RCNN, we consider an image as an input (Figure 7.) which passed through CNN for featuring images to get ROI of each. The process use ROI pooling layer to get it shaped again into fixed sizes so that to correlated with Fully-Connected (FC) layers.

Following are the steps taken for using Fast RCNN:

Step 1: Consider an image input.
Step 2: Pass this image to CNN for generating Region of Interest (RoI)
Step 3: Apply ROI pooling layer tall the regions to reshape this image, then pass it to fully connected network.
Step 4: On the top of fully connected network, Softmax (Linear) layer is used to generate classes as output. In parallel, a regression layer is used for bounding box coordinates to predict the classes.

From the above Figure 9, a bounding box is generated for each region to predict the targeted object. RCNN used region proposals at pixel levels into consequent CNN for detection purpose. Therefore, both the methods of RPN network through Selective Search (SS) algorithm and detection network get uncoupled, through, which if error occurred in SS network then it, may affect the detection network directly. So it is preferred that SS and detection network correlate with each other. However, training with RCNN is very expensive and slow as well because if we consider N images and 1000 features for extraction from them by using CNN, then it needs N*1000 CNN features which may make RCNN very slow. As around 50-60 seconds of time will take to have a prediction of individual images that make it a cumbersome model and practically not possible to host.

We can have all ROI by executing once for each image through Fast RCNN (Fast Region-based Convolutional Neural Network) because Fast RCNN uses RPN at feature map level instead pixel level. In Fast RCNN, we consider an image as an input (Figure 7.) which passed through CNN for featuring images to get ROI of each. The process use ROI pooling layer to get it shaped again into fixed sizes so that to correlated with Fully-Connected (FC) layers.

Following are the steps taken for using Fast RCNN:

Step 1: Consider an image input.
Step 2: Pass this image to CNN for generating Region of Interest (RoI)
Step 3: Apply ROI pooling layer tall the regions to reshape this image, then pass it to fully connected network.
Step 4: On the top of fully connected network, Softmax (Linear) layer is used to generate classes as output. In parallel, a regression layer is used for bounding box coordinates to predict the classes.

From the above Figure 11, it confirms that Fast RCNN resolute the issues of RCNN, as it passes single image instead to CNN and uses single model also for generating the boundary boxes for classification and extraction of features. Although, Fast RCNN uses SS algorithm for finding ROIs that makes the process very slow and time consuming. Although Fast RCNN detects an image in and around 2-3 seconds that may result had better compare to RCNN, yet it will not be effective for large real-time datasets. For such datasets, we do have another option of Faster RCNN (Faster Region-based Convolutional Neural Network) i.e., a modified version of Fast RCNN. In contrast, Faster RCNN (Figure 12) uses that considers image feature map as an input to produce the object proposals with objectness score as output (for reference, mentioned in Figure 1 as an object with objectness score).
Following are the steps taken for using Faster RCNN:

Step 1: Firstly, we take an input of image that returns a feature map while passing from CNN.

Step 2: For this Image Feature Map, we apply Region Proposal Network to generate the objectness score with its object proposal.

Step 3: To bring down the similar image, apply RoI pooling layer to all the regions and pass it to FC network.

Step 4: For class prediction, use the coordinates of boundary boxes by using linear regression layer.

Figure 12: Faster RCNN Framework

### III. RESULT AND DISCUSSION

By using camera-trap images, this practice focus only on deers without considering the images having more than one species. With this methodology, we classify and label the activities of captured deer. As given in Figure 13. (from left onwards), capture-1 represents sleeping activity, capture-2 represents grazing activity and capture-3 represents resting activity.

![Figure 13: Recognition of Deer Movements](image)

### IV. DISCUSSION

Our analytical method allowed us to make a clear and measured results of deer movements and on that basis only we may able to make a summary of Deep Learning methods below (Table 1.):

<table>
<thead>
<tr>
<th>Method</th>
<th>Premonition</th>
<th>Prophecy time / image</th>
<th>Restrictions</th>
</tr>
</thead>
<tbody>
<tr>
<td>CNN</td>
<td>Assign each regions into different classes by splitting into regions.</td>
<td>--</td>
<td>Very high computation time as it requires numerous regions.</td>
</tr>
<tr>
<td>Faster RCNN</td>
<td>Selects only bunch of regions, if find object the performs image extraction on or above 500 regions each image.</td>
<td>50-60 seconds</td>
<td>Since regions are passed separately from CNN, so have high computation time.</td>
</tr>
<tr>
<td>Faster RCNN</td>
<td>Images are passed only once through CNN (then extracts featured maps). For predictions, if passes only one image to CNN by using only 1 model instead, and then feature maps are extracted.</td>
<td>2-3 seconds</td>
<td>Computation time is quite high but selection of search make it 2-3 times.</td>
</tr>
<tr>
<td>Faster RCNN</td>
<td>Use region proposal network (RPN) instead some selective search.</td>
<td>0.15 seconds</td>
<td>Although, proposal of object takes time yet RPN makes it very faster.</td>
</tr>
</tbody>
</table>

Table 1: Measured Implementation of Deep Learning Methods Implications

### V. IMPLICATIONS

Our methodology necessitates minimum training set of 15-20 images which shorten the efforts of building a training set and code this perspective method is more convenient for wildlife researchers. Because of various systems working one after another, the further performance depends on how the system performed previously. All in all, encore from neoteric object detection algorithms such as YOLO (You Only Look Once), RatinaNet, SSD (Single Shot Multi-box Detector). These methods and algorithms are based on machine learning, imposing umpteen understanding of mathematical and deep learning frameworks. The work is a novel method for conferring the reliability for monitoring, detection and analyzing deer movements with auto-indexing from wildlife videos (Figure 1). The method handles occlusions, background clutters and most hidden deer behind vegetation. Experiments showed that accurate and robust detection of deer movements.

### VI. ACKNOWLEDGMENT

We thank to Amity Institute of Forestry and Wildlife - Amity University for assistance in the field. We specially thanks to all anonymous reviewers for helpful comments on earlier drafts as well.

### VII. CONFLICT OF INTEREST

The authors declare that there is no conflict of interest’s regarding the publication of this paper.

### REFERENCES


