IDENTIFYING INDIVIDUAL SPECIMENS AMONG SPECIES USING COMPUTER VISION

Mr. Aakash Ravindra Shinde
Department of Computer Science Engineering NBNSSOE, Pune, India

Ms. Praneeta Gopal Dumbre
Department of Computer Science Engineering NBNSSOE, Pune, India

Ms. Ruchira Kailas Borkar
Department of Computer Science Engineering NBNSSOE, Pune, India

Mr. Kshitij Hemant Patil
Department of Computer Science Engineering NBNSSOE, Pune, India

Dr. Amol V. Dhumane
(Guide), (Head of Department, Computer Science Engineering, NBNSSOE, Pune, India)

ABSTRACT

Earth provides shelter to more than 8.7 million species, when confronted with accelerating extinction rates of species during the last half of this century concerns for conservation of this ecological cycle is of utmost importance. Several methods are used for conservation of species and gather data for their sustenance, most of these methods utilises direct contact with the individual for their surveillance this results in a tedious process of handling the specimen and mounting devices on the individual and also causes these devices to stay on the individuals even after the devices are unable to function. Computer Vision and Artificial intelligence has shown promising results for the last decade for systems like facial recognition, object detection, etc. Camera trap methods were used to for tracking animals in a designated area but they rarely provide information about individual animal. We in this manuscript provide with a solution using Computer Vision for determining individual specimen among several species.

INTRODUCTION

In this manuscript we tried to provide with a proof of concept for identification of individual specimens amongst the species, for this we created our own dataset and trained models for identification of entities in image, further classification of the discovered entities and further trying to associate with the existing individual information to find the similarity and identify the individual. Camera Traps are being used for decades now for collecting information regarding the animals in a certain area such as sanctuaries or national parks, these Camera Traps even though extremely helpful for identifying various species in the vicinity but they lack the ability to identify an individual even after recursive interaction with the trap cameras. For tracking and identifying individual's RFID tags, GPS systems were used but they have several limitations, they needed to be mounted physically on the animals which includes animal handling and even the device may persist on the animal even after fulfilling its function, its next to impossible to tag and analyse all possible encountered entities also these mounting systems aren't designed for smaller species. Hence using the image-based information available we provide with a proof of concept for finding unique animal among the species using its body features and key points.

Computer Vision has been used extensively in fields related with zoological world, it has proved vital in identification of species from images and even determining the pose of animals, extensive research regarding the interaction of these fields has been presented in [1]. Experimentation on Computer Vision for identification individuals has been done but it's limited to animals with patterns present on their bodies like Tigers, Zebras and Jaguars but there is no solution provided that could be fit for both the patterned and pattern less animals [2]. The solution we proposed encompasses all these factors presented and still could prove effective. Before diving into the architecture and implementation procedure, following are few technologies utilised in this project which are helpful for better understanding the system.

1.1. Image Processing

A computer is unable to see and visualise image as humans to perceive them, for a computer and image is a collection of bits and bytes where each pixel is the brightness associate with it in a three-colour intensity formation. Image processing hurdles through series of steps like image reading in a certain format, image analysis over various spectrums, image manipulation and getting variations of image output. Examples of several digital image processing techniques are Independent Component Analysis, Anisotropic Diffusion, Linear Filtration, Pixilation, etc. [3].

1.2. Deep Neural Network (DNN)

Mimicking a process resembling inside of human brains of firing neurons an Artificial Neural Network (ANN) works. ANN comprises of multiple artificial neuron nodes; these neurons receive input data and perform simple operations on the received data and passing it to next neuron for further processing. Passing of these outputs to next neurons is dependent on activation functions. A deep neural network can be considered as stack of neural networks, comprising of several layers of networks among the neurons [4].

1.3. Convolutional Neural Network (CNN)

Considering DNN and CNN the neuron is not responsible for activation but neuron in CNN is a result of several convolutional operations carried out before getting activated for feature extraction or classification. CNN consists of multiple stages named convolution, pooling and non-linearity. These stages can be utilised several times to create a deep CNN. For a CNN input values are represented as a 2-Dimensional array on which convolutional operations are performed which are multiplication of these input variables with filter elements these filters can be weighted or bias filters. Pooling operation is performed for dimension reduction procedure, later nonlinearity like ReLU, tan-h or Sigmoid are used as activation function. Finally for classification a fully connected (FC) layer is added [4][5].

1.4. Mask RCNN

For solving instance segmentation problems in computer vision and machine learning, Mask RCNN a deep neural network is a prominently used for object detection i.e., Mask RCNN is capable of segregating various objects in an image or video file. Based on the training dataset provided the model provides with the object's boundary box, object's classes and masks. Mask RCNN works in two stages, in first stage depending on the input image it accesses the objects area and then generates the proposal of the region. In second stage, boundary boxes are further refined, objects class is also predicted while masking they are assessed in the stage one. Using a backbone structure these two stages are connected. Forcing different layers on a neural network for learning different scale feature is the advantages feature of Mask RCNN [6][7].

SYSTEM ARCHITECTURE

System architecture is designed with intensions of creating a recursively trainable system that once provides a satisfactory result, the results later when verified may be placed back into the training data for retraining of model to achieve better improvement. Hence the system relies on a ‘Verification Authority’ to check the results this helps to create a human verified recursive learning system. System gets its input images for processing from the uploaded images by the user into the system, before storing onto the system they are converted into the required format for the system. These formatted images are tested on the pretrained recognition systems which later provides results first for classification of the animal recognised in the image, with possibilities of having various animals in the same image and later find various key-points associated with the respectively classified animals.

Once the recognition system provides with its predicted results those results are reviewed by the authorities to be utilised by placing it in training dataset to retrain the model. The process of authentication is required to the point where model achieves a considerable accuracy as continuous recursion for learning in model improves its prediction results. Figure 1 below provides with the brief about the processing encapsulated architecture and provides a basic idea of data flow.

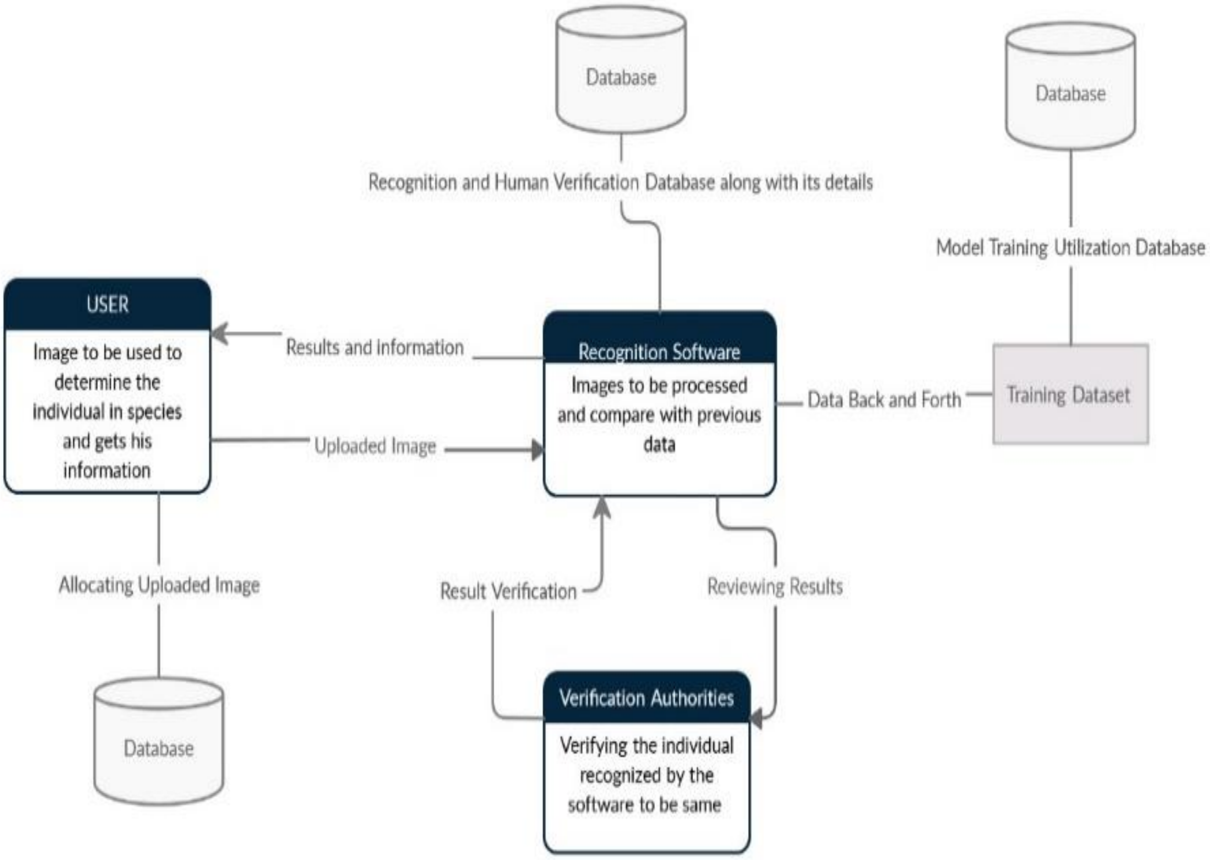
Database
Recognition and Human Verification Database along with its details
Database
Model Training Utilization Database
USER
Image to be used to determine the individual in species and gets his information
Results and information
Recognition Software
Images to be processed and compare with previous data
Data Back and Forth
Training Dataset
Uploaded Image
Allocating Uploaded Image
Result Verification
Reviewing Results
Database
Verification Authorities
Verifying the individual recognized by the software to be same

Figure 1.: System Architecture

RECOGNITION SYSTEM COMPONENTS

Recognition system in this architecture is the heart of the whole proposed system. Recognition system contains and is associated with multiple smaller systems performing smaller operations in series. Process in recognition system works in a flow with multiple steps, first the system collects the formatted image and tries to find objects or the number of entities present in the image and later classify them into the associated classes. For the sake of project feasibility and the available data we created five classes: Tiger, Dog, Cow, Horse and

Others. Others contains images of animals that do not belong to these four animals and once their class is available, they could be removed and the reinstated. This first stage concerns with animals' detection in the image, later in second stage after detection of the animal, boundaries of the animal are established and then the individuals in the image are saved with their unique id until they have found a recognition in the existing database. As there might be several animals in one image saving the cropped image individually is important for herd-based animals. Further in stage three various key-points related to the body structure of the animal are recognised and their points on the images are saved for further processing. In the fourth stage, these key-points are processed and compared with the pre-processed key-points of the animals existing in the database proposing a possibility with the animal having most common features related to the key-points.

Thus, this system provides a way to recognize individuals without having any physical contact with the individual or mounting any device on the entity. Once captured in an image an individuality is created just need to find in the database one with the closed feature with the new found image individual. Let's explore the various components associated with this system to further our understanding of this overall system, the components are as follows:

1.5.Dataset

Dataset is a quintessential part for any machine learning project, as without a well created, maintained and consistent data for training however efficient a machine learning model maybe but it would always produce faulty results. Hence, we created a dataset for the classes mentioned above, for each image in the corresponding dataset we manually labelled them for referencing the boundary boxes and key-points annotation. For the process of creating a dataset suitable for both animal classification and key-points detection we used an free, open source, web-based image and video annotation tool named Computer Vision Annotation Tool (CVAT) used prominently for labelling data for computer vision algorithms [8]. CVAT has many powerful features, including interpolation of shapes between key frames, semi-automatic annotation using Deep Learning models, shortcuts for most critical actions, a dashboard with a list of annotation projects and tasks, LDAP and basic access authentication, etc. CVAT is written mainly in TypeScript, React, Ant Design, CSS, Python, and Django. It is distributed under the MIT License, and its source code is available on GitHub [8].

We researched several available datasets that could be appropriate for working of the system, one of which is Dataset used for Cross-domain adaptations for animal pose estimation by et al. [9].It is a dataset which provides animal pose annotations for five categories which are dog, cat, cow, sheep and horse. It also contains a bounding box annotation for other 7 animal categories. There is a total of 20 key-points annotated in images of this dataset. Even though this dataset works great for pose estimation it lacks with the ability to identify each unique key-points on the animal, it contains points that are far too generalised. ATRW (Amur Tiger Re-identification in the Wild) dataset is an another reviewed non-commercial/research purposed dataset made on the amur tiger and leopard conservation programme but it is only limited for patterned animals and has no way to determine key-points [2]. To overcome all these setbacks from these datasets we created a dataset named AUKD (Animal and Unique Key-point Detection) dataset, that contains label of animals, boundary box annotations named as xtl (x coordinate of top left), ytl (y coordinate of top left), xbr (x coordinate of bottom left) and ybr (y coordinate of bottom right) denoting two diagonal points of the boundary rectangle. We also named each unique key-points for each of animal as specified in the Table 1. below.

Table 1. List of labels for each key-points specific to animals

Dog	**Horse**	**Tiger**	**Cow**
R_F_Paw/Foot	R_F_Paw/Foot	R_F_Paw/Foot	R_F_Paw/Foot
L_F_Paw/Foot	L_F_Paw/Foot	L_F_Paw/Foot	L_F_Paw/Foot
R_F_Shoulder	R_F_Shoulder	R_F_Shoulder	R_F_Shoulder
L_F_Shoulder	L_F_Shoulder	L_F_Shoulder	L_F_Shoulder
L_B_Stifle	L_B_Stifle	L_B_Stifle	L_B_Stifle
R_B_Stifle	R_B_Stifle	R_B_Stifle	R_B_Stifle
L_Eye	L_Eye	L_Eye	L_Eye
R_Eye	R_Eye	R_Eye	R_Eye
L_B_Hock	L_B_Hock	L_B_Hock	L_B_Hock
R_B_Hock	R_B_Hock	R_B_Hock	R_B_Hock
R_F_Elbow	R_F_Elbow	R_F_Elbow	R_F_Elbow
L_F_Elbow	L_F_Elbow	L_F_Elbow	L_F_Elbow
R_F_Knee	R_F_Knee	R_F_Knee	R_F_Knee
L_F_Knee	L_F_Knee	L_F_Knee	L_F_Knee
R_B_Paw/Foot	R_B_Paw/Foot	R_B_Paw/Foot	R_B_Paw/Foot
L_B_Paw/Foot	L_B_Paw/Foot	L_B_Paw/Foot	L_B_Paw/Foot
Nose	Nose	Nose	Nose
Neck	Neck	Neck	Neck
Withers	Withers	Withers	Withers
Tail_Base	Tail_Base	Tail_Base	L_Horn_Tip
			R_Horn_Tip
			L_Horn_Base
			R_Horn_Base
			Tail_Base

These mentioned points are the once that even when an animal is in motion or in different positions could prove helpful in determining stiff structure of the animal. We have even labelled them carefully based on the perspective of the animal as mentioned in the Figure 2 below for left and right. These points provide us with the distance among the skeleton of the animal which when further analysed provides with the unique structure to compared to the others in the same species. Depending upon the species introduced to the system new key-points can be considered and added onto the system for further processing. We used 'xml' file format to store this information along with the image name and the labels associated.

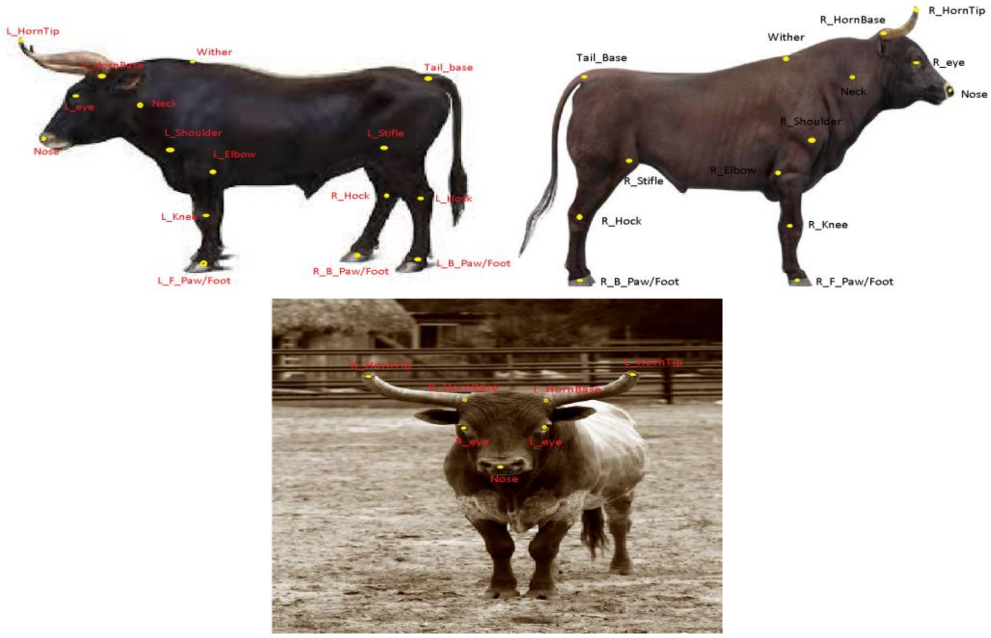

Figure 2.: Labelled example of animals

1.6. Classification and Boundaries

After collecting image from the user and transforming it into specified format next part of recognition software is identifying the animals present in the images this process of identification is carried out in phase of classification for detecting object entities and classifying them. Using the previously generated dataset and Mask RCNN model discussed earlier we devised a classification system that provides with the classified label and the boundary box for the animals recognised in the images. We provided the Mask RCNN model with training and testing dataset containing around 1167 manually labelled images each. Mask RCNN has proved its worth for multiple object detection base projects, we created total six classes to classify those are the five labels discussed prior and 'background'.

Mask RCNN was set at learning rate of 0.006 with skipping the detection with confidence less than 90 percent while training along with max ground truth instance to 10. Resnet101was the Backbone used with strides of 4,8,16,32,64 with pool size of 7. This model was trained for 5 epochs with each consisting of 131 steps per epoch. While training the model was consistently evaluated for loss, rpn_loss, rpn_bbox_loss, mrcnn_class_loss, mrcnn_bbox_loss and mrcnn_mask_loss, based on these constraints best models were evaluated and saved as prediction model to be called whenever required for prediction. The Mask RCNN was assigned one GPU per image and hence required extensive amount of time to train the models on the existing system. By considering few of the images in the existing dataset for testing purpose by recursive testing we were able to estimate around 95% accuracy regarding classification of the and generation of the boundary box. Later these classified images were separately saved into their respective databases for further determination of key-points.

1.7. Key-point Detection

After segregating the images into their respective databases, they are further analysed using a convolutional neural network for detecting key-points associated with the specific species. Key-points are an essential factor for this system in determining the individuality of an individual, as specified prior about the dataset containing more than 800 images with specific key-points the CNN was trained. This system requires integration and transformation of two files into a single easily processible file hence data frame creation technique using pandas was implemented.

From the xml file containing the x, y coordinates of the key-points are collected and saved in a data frame with their associated filenames and labels creating a 2-dimensional structured tabular data easier for processing. This creates a list with 48 variables that are to be determined based on the existing images. Later image processing is performed which is grey scaling. Grey Scaling is a technique utilised so that the computer can manipulate by using the process of 'converting continuous-tone' which converts coloured images into black and white images not only reducing the size of the images but also helpful for the model to recognise specific points associated with the animal. For processing the image converting it into an array is favourable for option as it helps in convolutional network to process it.

After the grey scaling the image size was reduced but still not enough for further processing hence the images are further reduced to 96×96-dimensional size images. Later these images are flattened and resized into one dimensional array so that they can be given as an input to CNN. Finally, both the data frames containing the information of the key-points and one-dimensional array are combined using their filenames specified inside the xml file and the actual filenames. These are later split into train and test set for 80-20 split, with datapoint shape described as (number of images, x dimension of image, y dimension of image, dimensional size of array) and labels shape described as (number of images, number of labels).

Further this processed data is made available to the CNN model. CNN model is designed with each having batch size of 64 and with iterations of over 500 epochs. CNN model consisted of 5 layers (16,32,64,128 and

256) with the activation function being ReLU with a sequential fashion of processing through these layers. Finally at the end a Dense layer is added with a dropout rate of 0.2. With 500 epochs the model is iterated that many times where each are saved on a checkpoint model file and replace with the model with least val_loss and better accuracy. For optimising of the model during compiling 'adam' optimiser is used with loss metric being 'mean squared error'. Finally, the best model is saved with a corresponding filename and used for testing. Accuracy of overall 35% was achieved for determining those key-points this lesser accuracy is due to lack of available data (less than 208 images with annotations) and human time required to create data of key-points for one entity (15 min at least for manual labelling). For testing of external input image, the images undergo similar processing done before on training images and then applied to the model, which further returns an array with values associated with the key-points.

1.8. Identification of Individuals

Reason behind considering specific key-points mentioned above was that those key-points help to measure multiple firm skeletal points that are firm even if the entity is in motion or in any positions. So once getting the key-points from the previous systems are acquired Euclidean distance among specific parts are calculated with their pixel values. Several distances like the Left eye to Right eye, Nose to Left eye, Nose to Right eye, Nose to Neck and Horn Tips to Horn Bases help the system estimate the size or outline the skull while skeletal distances like Tail Base to Wither, Elbows to Knees, Shoulders to Knees, Stifles to Hocks, Knees to Front Paws/Feet and Hocks to Rear Paws/Feet help indicate the size or outline of the animal's torso. Even if we get the distance, we cannot surely pinpoint the actual size of the skeletal length as there are multiple factors that can affect is first of all the position of the animal in the image the depth of perception surely affects the pixel length and is incomparable. Even if we use an estimate with the average for measuring its skeletal length it still might provide with faulty results as it hampers the characteristic of individuality by averaging it.

Method to overcome this situation was we considered ratios of each distance with every other distance we can get from the image. Even when the same entity is positioned at different depth the ratio of the recorded distances won't vary by much. This solution persists over the perspective problem and hence solves the problem of determining features to find an individual among the herd. Another advantage of ratioing is that it provides with even more comparing points even if certain key-points aren't present in the actual image. One of most important things was to get a comparative data of the individual to find the individual in the database. Hence, we traversed through the xml files getting the points form the labelled images finding their ratioed distances and adding that data to a comma separated file for future comparison.

During this procedure we had assigned key-points that are not present or indistinguishable a value of '0.0' which when considered for ratio gives a divide by zero error, this is handled by assigning extremely high value so that it turns out to be obsolete when comparing. Comparison is done by subtracting comparing array of values with the compared array of values, when value lies between -1 to 1, we consider it as a potential match and hence the distanced ratio is marked with 1 point else its 0, top 5 with most number of points are considered as the major candidates that might be the individual in the image.

Finally, the output from the recognition systems of possible individual candidates is evaluated by the verification authority and possible match is added back into the system with the same unique identification value adding another parameterized image for later comparison, even if there is no valid pair then it can be provided with new unique identification value and stored as a new entity. This recursive action helps is creating a robust model at the point when the verification authority could be removed and a whole model could be created that would not require any human interference. Once we get the unique identification of an entity is acquired multiple important information could be associated with the value which could help create a universal database with information of various individual entities accelerating the process of zoological research even helpful for keeping data of migratory species. We have even considered this system for younglings as most of the younglings grow at a steady rate with the same ratio among their body parts except for some unnatural growth, hence we can even have the same data even if the entity grows this structure would hold the same ratio throughout his life.

RESULTS

For this whole process we used an Intel i7 10th generation processor system with 8GB RAM and GPU of size 2GB, with an internal SSD to store the data. This system takes tremendous amount of time to train a system with large amount of data even with a dataset with only 1167 image it takes around 3 hours training one epoch of classification algorithm. As images were highly compressed for training the CNN model comparatively it took lesser time but this affected the accuracy expensively as the information regarding the points was also reduced compensating with the process. As the initial system failed the comparison system was affected even though it’s a linear arithmetic system without predictions but comparisons.

Even with all these drawbacks of the system presented this system provides with a proof of concept that doesn’t involve any physical interaction with animal’s information can be acquired to find their unique identity and monitor them around the globe. For this system to be highly function a lot of manual work needs to be put in order to create a well verse dataset that’s is labelled with similar convention of the dataset used above. Classification would not be hampered for individual entities in an image but they would surely need a lot of data for multiple entities is an image and of different species. Manual task of determining key-points for dataset creation could be changed from manual input to monitor by just getting output from the current system and updating the wrong results and adding the results back into training model for attaining an accurate model. As we were trying to provide a proof of concept our basic setup wasn’t able to work as well as a dedicated system might work as majorly suggested for most of the computer vision applications.

Focusing on obtaining multiple images of the same entity might provide this system might prove helpful for training the CNN model more efficiently and might reduce its loss valuation as the model has more value to relate to a same entity finding positioning the key-points even better value. More structured data would prove more helpful of the entities for creating similar dataset with extensively available animals of same species, later these species characterisation into subspecies could even prove helpful for better individual estimation. Mask RCNN being exceptional for object detection based on the available image lacks at few places. Mask RCNN is an open-source community-based software it hasn’t been updated for latest systems and lacks even further as these systems upgrade, considering that Mask RCNN works on systems with TensorFlow below 1.15 and Python below 3.7.0 and as the systems are using TensorFlow above 2.0 after update and Python 3.8 above as 3.9 is the latest version it might not work on several updated systems as they would be required to be downgraded.

FUTURE SCOPE

Although results don’t seem to be as fascinating but from proof of concept point the system holds its ground and once upgrade it might be able to work in field with trap cameras in place in wild. Trap cameras have been successful for species identification and patterned animals’ recognition [10][11] but with this system mounted those manual efforts would be reduced even further and more information could be obtained from same existing monitoring systems. Tourist might even turn into a source of data for animal conservation as images taken from the visitors when applied to the system might even help position the entity based on the location of the photo taken. This could even work by bringing more environmental tourism with this kind of interactive system benefitting the sanctuaries and national forests. With a centralised system tracking of migratory species could be done at an extensive level as images from any places could show potential individual tracking with the location of images taken.

With a global integration of this system, we can eliminate the use of systems of mounted systems as these individuals could still be tracked without any RFID tags or GPS, these devices even though might not seem harmful for animals but are potentially trash on the lands as they still persist after the decreased animal. These mounted devices aren't able to scale with the number of individuals like a large herd of wilder beasts. This system also eliminates the tedious task of tranquilizing the animals and then mounting these devices and gets rid of those physical hassle. Considering all these factors this system might provide a better option to current ways of zoological studies just it needs to be implemented on a larger scale and provide it with a extensively well working physical hardware along with the extensive amount of data for processing.

CONCLUSION

Using the technological prowess presented by the computer vision we presented a solution for identification of individual entities among species. We were able to present a proof of concept by implementing the idea presented along with a dataset example that might provide with an outlook on the way a new dataset for zoological research. We used several deep learning models for classification of the entities found in the images and finding their key-points so that these points could be further processed in order to find unique attributes stating the entities individuality. We even presented an idea on ways one can utilise the structural information available in the image of an animal regardless of its depth perception or the method of visualisation. Even though the system lacks in prediction accuracy but this instantiates the lack of availability of good data, well versed dataset and the lack of hardware utilised. Still this system provides a proof that this theoretical approach has a practical use.

This system shows great potential not only form a research perspective but from an approach to educate and involve more people in the ways of educating people to conserve fauna in an interesting manner. With more improvements in the hardware, more availability of data and inclusion of people on a vast scale this system would help in conservation in an unprecedented manner.

REFERENCES

1) A. R. Shinde, K. H. Patil, P. G. Dumbre, R. K. Borkar, and D. A. V Dhumane, "A Survey on Augmentation of Computer Vision and Image Processing Techniques with Zoological Research for Anatomical and Speciating Speculations," *Int. Res. J. Eng. Technol.*, 2021, Accessed: Jun. 08, 2021. [Online]. Available: www.irjet.net.
2) G. S. Cheema and S. Anand, "Automatic Detection and Recognition of Individuals in Patterned Species," *Lect. Notes Comput. Sci. (including Subser. Lect. Notes Artif. Intell. Lect. Notes Bioinformatics)*, vol. 10536 LNAI, pp. 27–38, 2017, doi: 10.1007/978-3-319-71273-4_3.
3) "Image Recognition and Image Processing Techniques | by Adoriasoft | Medium." https://medium.com/@Adoriasoft/image-recognition-and-image-processing-techniques-fe3d35d58919 (accessed Jun. 07, 2021).
4) H. S. Das and P. Roy, "A deep dive into deep learning techniques for solving spoken language identification problems," in *Intelligent Speech Signal Processing*, Elsevier, 2019, pp. 81–100.
5) S. Albawi, T. A. Mohammed, and S. Al-Zawi, "Understanding of a convolutional neural network," in *Proceedings of 2017 International Conference on Engineering and Technology, ICET 2017*, Mar. 2018, vol. 2018-January, pp. 1–6, doi: 10.1109/ICEngTechnol.2017.8308186.
6) "Simple Understanding of Mask RCNN | by Xiang Zhang | Medium." https://alittlepain833.medium.com/simple-understanding-of-mask-rcnn-134b5b330e95 (accessed Jun. 07,

2021).

7) K. He, G. Gkioxari, P. Dollár, and R. Girshick, “Mask R-CNN,” *IEEE Trans. Pattern Anal. Mach. Intell.*, vol. 42, no. 2, pp. 386–397, Feb. 2020, doi: 10.1109/TPAMI.2018.2844175.

8) “New Computer Vision Tool Accelerates Annotation of Digital Images and...” https://www.intel.com/content/www/us/en/artificial-intelligence/posts/introducing-cvat.html (accessed Jun. 08, 2021).

9) J. Cao, H. Tang, H. S. Fang, X. Shen, C. Lu, and Y. W. Tai, “Cross-Domain adaptation for animal pose estimation,” *arXiv*, pp. 1–14, 2019.

10) A. Swanson, M. Kosmala, C. Lintott, R. Simpson, A. Smith, and C. Packer, “Snapshot Serengeti, high-frequency annotated camera trap images of 40 mammalian species in an African savanna,” *Sci. Data*, vol. 2, Jun. 2015, doi: 10.1038/sdata.2015.26.

11) M. S. Norouzzadeh *et al.*, “Automatically identifying, counting, and describing wild animals in camera-trap images with deep learning,” *Proc. Natl. Acad. Sci. U. S. A.*, vol. 115, no. 25, pp. E5716–E5725, 2018, doi: 10.1073/pnas.1719367115.