# Mammalian Species Detection Using a Cascade of Unet and SqueezeNet

Michael Njeru
Electrical Engineering Department
Pan African University Institute for Basic Sciences Technology and Innovation
Juja, Kenya
michaelmutwiri@gmail.com

Ciira Maina
Department of Electrical and Electronic Engineering
Dedan Kimathi University of technology
Nyeri, Kenya
ciira.maina@dkut.ac.ke

Kibet Langat
Telecommunication and Information Engineering Department
Jomo Kenyatta University of Agriculture and Technology
Juja, Kenya
kibetlp@jkuat.ac.ke

2021 IEEE AFRICON | 978-1-6654-1984-0/21/$31.00 ©2021 IEEE | DOI: 10.1109/AFRICON51333.2021.9570950

***Abstract*— Monitoring of wild animals has taken different approaches with an aim to provide vital information used in animal protection in their natural habitats. To recognize animal species without human trackers requires machine learning models that extract specie's features from an image. This project proposes a method of counting animals in an image and specifying the species of each animal using Unet and a variant of the SqueezeNet model. To train the Unet model, images and corresponding masks are used as the training data. Different optimizers are applied to each model. During inference, Unet outputs a binary mask with ones where an animal is detected and zeros elsewhere. SqueezeNet model is trained with images corresponding to six classes: bushbuck, impala, llama, warthog, waterbuck, and zebra. Three variants of the SqueezeNet model have been trained. The first contains the original backbone while the other two have the original backbone with an additional fire module. In one model the Fire module is similar to the Fire modules of the original backbone while in the other model, the extra fire module contains batch normalization layers. The trained models show that Unet trained with Nadam optimizer achieves the highest dice coefficient while the SqueezeNet with an extra Fire module containing batch norm layers and RMSprop optimizer achieves the highest training accuracy. The combined system containing the two models takes an image and outputs the image with bounding boxes around each animal and the corresponding animal species. The system achieves both counting and recognition of the species for each image placed at the input.**

***Keywords—Animals, Unet, SqueezeNet***

## I. INTRODUCTION

Wildlife conservation is an ongoing process throughout the world. The wild animal population has declined by about 60% in the last 40 years [1]. Most of the mammalian wildlife is being protected in limited areas in form of animal parks, conservancies, and sanctuaries. One of the key aspects of wildlife conservation in these areas is monitoring the availability and movement of animals in their natural habitat. Both human trackers and technological methods have been used to determine the movement and presence of animal species in a given area. Human tracking is quickly being eliminated due to the risks such as being attacked by animals and bad terrain in some areas. The use of sensors, the Internet of Things (IoT), and wireless sensor networks have been applied in many ways to achieve the basic function of monitoring animals[2], [3].

The use of radio-frequency identification (RFID) technology has been widely used both for domestic and wild animals[4][5]. This technology requires the attachment of RFID tags on animals which is not easy for some types of animals. In a wildlife setup, only samples of animals are attached with the device which gives the general idea of the overall whereabouts of the groups of animals. The devices have been integrated with wireless sensor networks and IoT to eliminate the use of humans with radio receivers. This technology can provide information about the movement of animals. In combination with GPS tracking and other sensors, migrations patterns are deduced from recovered data[6][7]. Animals tend to form groups that merge and reform over time. While this method is very vital for endangered species, the use of RFID for tracking solitary animals or a species with a large population requires many animals to be attached with the devices to increase the monitoring accuracy of the whole population. For each species, a different RFID marker is required. This method requires that once a tagged animal is dead, the tag be removed from the carcass and utilized in another animal else it is lost.

The use of camera traps can observe all species that are within the field of view at once. Camera traps have been widely deployed where image data is collected manually from cameras or images transmitted wirelessly to a storage center [8][9][10]. This method requires human labor or wide bandwidth for transmitting images. This approach is however very essential where images are required for display. Automatic detection and classification of animals on the captured images has been widely approached using machine learning models. Deep convolutional neural networks (CNN) have shown improved accuracy in the recognition of objects. Different CNN models such as SqueezeNet[11], VGGNet[12], ResNet [13], and Mask R-CNN [14] have been trained to perform object recognition. The training process requires a large number of images and a high computer power processor to achieve high accuracy. The applications of CNN models have been applied in animals detections with appreciable accuracy, however, the most accurate models rely on large training data and generate large model files.

This paper utilizes two machine learning models working together to recognize animal species and count them from images. The combined models form a system that can recognize and count animal species in an image with appreciable accuracy.

The rest of this paper is organized as follows: The literature review is represented in section 2. The methodology used is outlined in section 3 while the results and their discussion is contained in section 4. Section 5 is the last and contains the conclusions and ideas on future work.

## II. LITERATURE REVIEW

Different machine learning models have been developed. The highest accuracy achievable by a model at least depends

on the size of the training data available and the design of the model itself. The applications of a model in a real system depend on the size of the model, computation requirements, and the minimum accuracy that can be accepted. In the process of monitoring animals, wireless sensor networks have been used as a means of capturing the images which are transmitted to a computer equipped with machine learning models for classification.

Research in [15] developed an animal detection system based on deep learning that could identify, count, and describe the actions of animals in a camera trap image. The training and testing data set was obtained from the database of Serengeti National Park and only images containing one species were selected. Deep Neural Networks (DNN) that were trained and compared included AlexNet, NiN, VGG, GoogLeNet, and ResNet. The complete system performed a sequence of four tasks: detecting images that contain animals; identifying species; counting animals and additional attributes. One of their most important results is eliminating human labor in removing images without animals. These models generate huge file sizes that in most cases are considered too large for implementation on embedded processors.

In systems with low memory and low computational power, lightweight models are required for image classification. MobileNet[16] was developed with this goal. Variants of MobileNet has been developed such as Dense-MobileNet[17], Dilated-MobileNet[18], MobileNetV2[19] and MobileNetV3[20]. All these versions of MobileNet aim for high accuracy while maintaining simplicity and fewer parameters than VGGNet, GoogLeNet, and ResNet. In [21], MobileNetv2 is used for the prediction and classification of x-ray images with lung diseases and achieved an accuracy of 95%.

A much smaller convolutional neural network is SqueezeNet[22] which has more than 50 times fewer parameters than the other state of art architectures. This model has been envisioned for real-time applications. In [23] SqueezeNet is chosen for its small size and reduce computational complexity while maintaining high accuracy. Comparative research studies involving SqueezeNet among other modern CNN architectures such as in [24][23][25] show the accuracy of SqueezeNet is comparable to the most recent CNN architectures. Transfer learning with SqueezeNet has been used in the detection of abnormality in chest x-ray images in [26] using very few images and achieves over 90% accuracy.

Animal monitoring has taken the approach of cameras in the field and transfer of images over the wireless sensor network in some cases such as [10]. In this case, the terminal nodes are equipped with high-resolution cameras and a wireless image sensor network established using ZigBee modules. Processing images at the node level have been done in [27] to count people and transfer this number to a website for display.

## III. METHODOLOGY

Both Unet and SqueezeNet are applied together to achieve species recognition and counting. Unet is used to generate masks around each animal. Bounding boxes around each animal help to count the number of animals in each image. SqueezeNet is used to recognize the species of each animal masked. The project is implemented in the following stages: data preparation, model training, and forming a cascade of the two models into a complete working system.

### *A. Data preparation*

Training data was downloaded from Labeled Information Library of Alexandria: Biology and Conservation (LILA-BC) datasets. The specific dataset downloaded are Great Zebra and Giraffe Count and ID[29], Snapshot Enonkishu[30], Snapshot Mountain Zebra[31], Snapshot Kruger[32], Snapshot Serengeti[33], WCS Camera Traps[34]. Only 5 species were of interest from these datasets. There are bushbuck, impala, warthog, waterbuck, and zebra. Due to huge dataset files, only images of these species were downloaded using a program provided by LILA BC. The downloaded images were sorted manually to remove empty shots and night images. Cameras used to capture images in the above datasets take three shorts when triggered and labels are provided per shot thus leading to some empty shots if the animal was on the run. Llama is the sixth species in this project and the images were sourced from freely usable images at the Unsplash site [35].

VGG Image Annotator (VIA) was used to prepare masks of the gathered dataset [36]. For the Unet, masks are required during training. To train the SqueezeNet, images were resized to the size of input of the model, that is, 227 by 227.

### *B. Models and training*

Two models, Unet and SqueezeNet were trained separately. The original Unet[37] shown in figure 1 is modified by using MobileNetv2 as the encoder with imagenet pre-trained weights. The training dataset consists of images and their corresponding masks. The output is a binary mask of the animals. The training process requires setting the learning rate, the number of epochs as well as monitoring validation loss and Dice coefficient. Learning rate is a parameter that determines the step size when updating the weights of the model. An epoch is the training process in which all data is passed and used exactly once. Dice coefficient is a measure of the relative overlap between the ground truth masks and the predicted masks. Validation loss is a measure of the error of the model calculated using the validation dataset. During training, the learning rate reduces when validation loss is not improving, and training stops when validation loss does not improve in 10 epochs. The model was trained with two different optimizers, that is, Adam and Nadam. Dice coefficient was monitored during the training process. The dice coefficient is given by the following equation

$$Dsc = 2|A \cap B| / (|A| + |B|) \quad (1)$$

Where dsc is dice coefficient, A is the area of the original mask, and B is the area of the predicted mask.

SqueezeNet architecture was modified by adding an extra fire module on top of the original architecture before the classifications layers as shown in figure 2. Three model architectures were built, (i) model with the original backbone, (ii) model with original backbone plus one extra fire module, and (iii) model with original backbone plus one extra fire module with batch normalizations layers as shown in figure 2. Each of the three models is trained using three different optimizers (Adam, RMSprop, and stochastic gradient descent (SGD)). In each case, the learning rate (Lr) was reduced exponentially according to the equation below.

$$Lr = Lr * \exp(0.1) \quad (2)$$

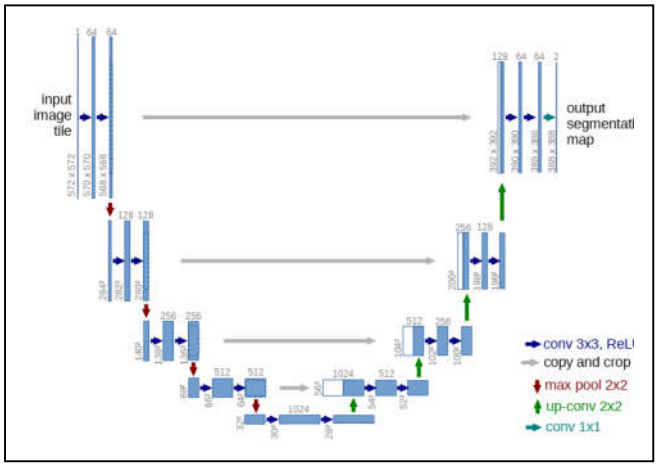


Fig. 1. Original Unet architecture with blue boxes representing mult-channel features maps and blue boxes correspond to copied feature maps

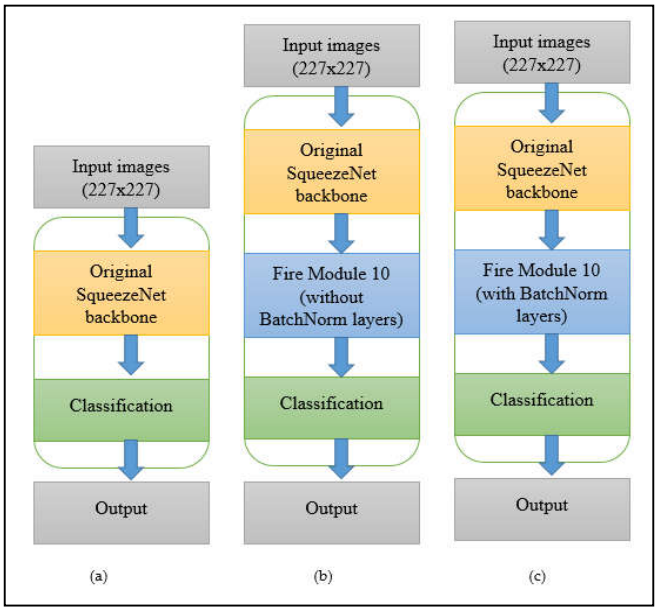


Fig. 2. The three different SqueezeNet architectures were built for training. (a) The original backbone architecture (b) The original backbone architecture with extra fire module without batchnorm layers (c) The original backbone architecture with extra fire module containing batchnorm layers.

### C. *Cascade of the two models*

A program was written to bring together the best Unet model and SqueezeNet model. The program takes an image as input then outputs the number of animals and species of each animal. The flow chart in figure 3 shows the stages of the combined system.

A resized image is fed to the Unet which outputs a binary mask. The mask is resized back to the size of the original image which is further used to generate bounding boxes around each animal detected. Regions corresponding to each bounding box are extracted and fed to the SqueezeNet model one at a time. Each region is considered as an animal and its species is determined from the SqueezeNet model output. The number of bounding boxes represents the total count of the animals.

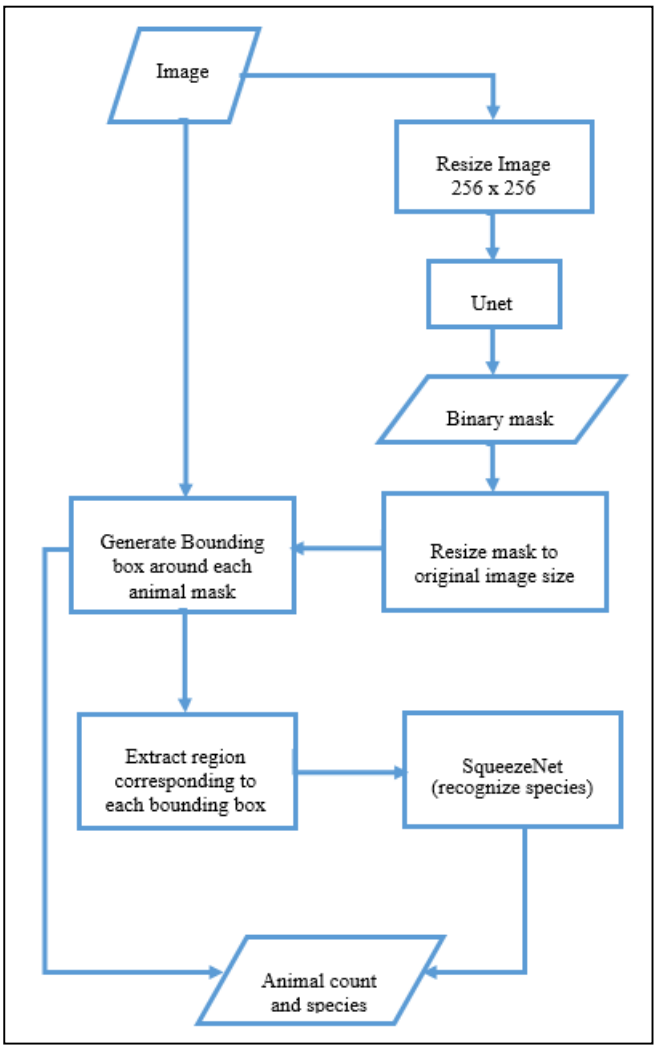


Fig. 3. Flow chart of the combined system for animal recognition and counting.

## IV. RESULTS

The trained models were deployed into the Raspberry Pi at the sensor node level and the detection accuracy of the images captured in the field compared to the accuracies obtained during training.

### A. *Model training results*

Unet model trained using Adam and Nadam optimizers achieved dice coefficient values shown in table 1. Nadam optimizer achieves a higher dice coefficient and therefore is the best choice. Three variants of the SqueezeNet model were trained using Adam, RMSprop, and stochastic gradient descent (SGD) as summarized in table 2. RMSprop achieves the highest accuracy and therefore becomes the best choice.

TABLE I. THIS IS A TABLE OF DICE COEFFICIENTS ACHIEVED WITH DIFFERENT OPTIMIZERS OF UNET

| **Model** | **Optimizers** | |
|---|---|---|
| | ***Adam*** | ***Nadam*** |
| Unet (encoder = pretrainedMobileNetV2) | 0.9615 | 0.9655 |

TABLE II. THIS IS A TABLE OF ACCURACIES OF THE THREE SQUEEZENET MODELS TRAINED USUING DIFFERENT OPTIMIZERS.

| Model | Optimizers | | |
|---|---|---|---|
| | *Adam* | *RMSprop* | *SGD* |
| Original SqueezeNet model | 95.31% | 96.45% | 42.34% |
| Original SqueezeNet model + extra fire module without Batch Norm layers | 93.39% | 96.12% | 49.38% |
| Original SqueezeNet model + extra fire module with Batch Norm layers | 97.85% | 97.96% | 67.82% |

Unet model trained with Nadam optimizer and version of the SqueezeNet model with extra fire module containing batch normalization layers and trained using RMSprop optimizer are selected for the combined system.

### B. *Combined model testing*

An image is passed through the system and the first stage generates a mask. Figure 4 shows an image and the masked produced by the first stage containing the Unet model.

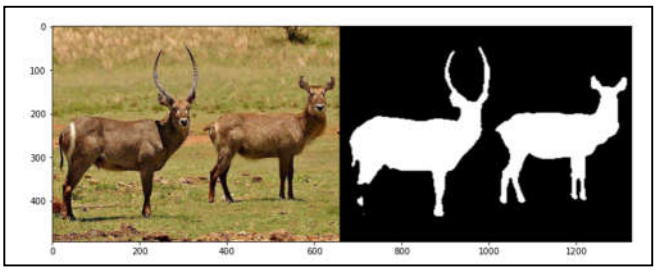

Fig. 4. Image of two waterbucks and the corresponding generated masks by the Unet.

After the mask is generated by the Unet, it is used to create a bounding box around each detection. The box dimensions are used to extract a region of the image for each detection. Each extracted image is passed through the SqueezeNet for recognition. The output is combined as shown in figure 5.

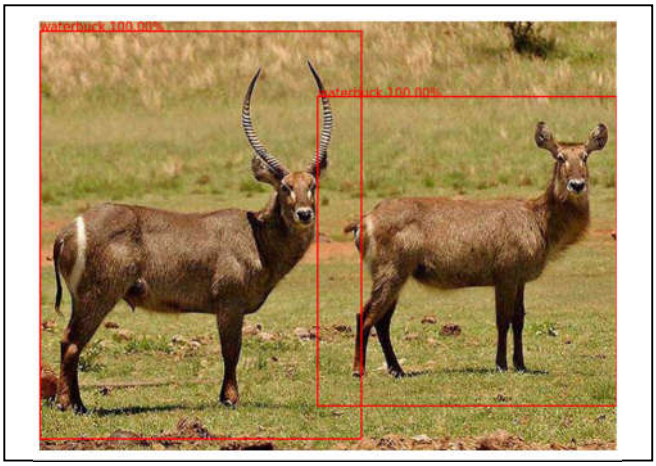

Fig. 5. Predictions made by the SqueezeNet and bounding boxes around each animals.

The Unet falsely detects small non-animal objects. These lead to regions containing no animal or just part of the animal being passed to the SqueezeNet for recognition. These results in errors. An example is shown in figures 6 and 7. To minimize these errors, a region smaller than 100 pixels is ignored. The number of pixels indicating the smallest acceptable region can be adjusted based on experimentation of the nature of images targeted.

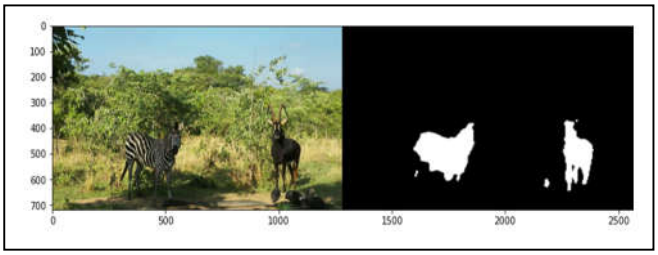

Fig. 6. An image with mask containing erroneous objects detected.

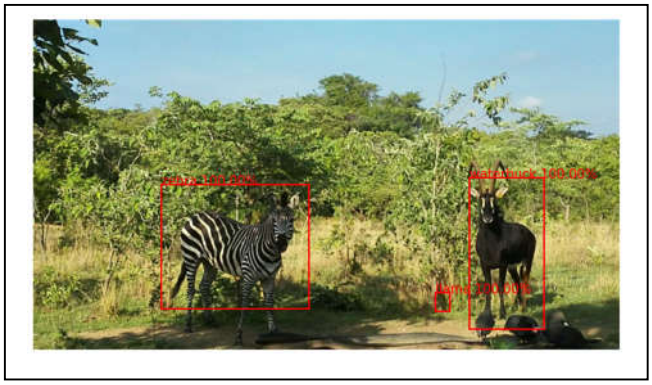

Fig. 7. Detection results showing a small non-animal objects falsely detected as animals.

## V. CONCLUSION AND FUTURE IMPLEMENTATION

Simple Convolutional neural networks can be combined to achieve more functions while maintaining simplicity and reduced computational power. Transfer learning with Unet and SqueezeNet models achieves high accuracy with a much smaller training dataset. SqueezeNet accuracy can be improved by adding an extract fire module on top of the original backbone.

A proposed future implementation of the complete system will contain a Raspberry Pi 4B, a motion sensor, a camera, and a T-beam TTGO T-Beam board. The motion sensor will be a 120 Degree (PIR) Pyroelectric Infrared with a reaction distance of 6m to 7m. The camera will be a Raspberry Pi Camera V2.1 8MP. The TTGO T-Beam board v1.0 integrates LoRaWAN communication, Wi-Fi, Bluetooth Low Energy (BLE), and GPS module [28]. A motion sensor will be used to trigger the camera which will capture three shots for every trigger. The images will be passed through the system illustrated in figure 3. The results will be uploaded to the 'The THINGS NETWORK' (TTN) console.


## ACKNOWLEDGMENT

First, I would like to acknowledge the great support of my supervisors, Dr. Ciira Maina and Dr. Kibet Langat, for their advice and assistance in making this project a success. My second appreciation goes to the financial sponsor of this research, the African Union (AU), for financial assistance through the Pan African University Institute for Basic Sciences Technology and Innovation (PAUSTI). Finally, I would like to say a big thank you to my family and colleagues at the Pan African University.

## REFERENCES


[1] [1] WWF, *Living Planet Report - 2018: Aiming higher.*, vol. 26, no. 04. 2018.

[2] [2] A. Prosekov, A. Kuznetsov, A. Rada, and S. Ivanova, "Methods for monitoring large terrestrial animals in the wild," *Forests*, vol. 11, no. 8, pp. 1–12, 2020.

[3] [3] D. J. Ingram, D. Willcox, and D. W. S. Challender, "Evaluation of the application of methods used to detect and monitor selected mammalian taxa to pangolin monitoring," *Glob. Ecol. Conserv.*, vol. 18, p. e00632, 2019.

[4] [4] V. M. Anu, M. I. Deepika, and L. M. Gladance, "Animal identification and data management using RFID technology," *Proc. 2015 - IEEE Int. Conf. Innov. Inf. Comput. Technol. ICIICT 2015*, 2016.

[5] [5] A. S. Voulodimos, C. Z. Patrikakis, A. B. Sideridis, V. A. Ntafis, and E. M. Xylouri, "A complete farm management system based on animal identification using RFID technology," *Comput. Electron. Agric.*, vol. 70, no. 2, pp. 380–388, 2010.

[6] [6] V. R. Jain, R. Bagree, A. Kumar, and P. Ranjan, "wildCENSE: GPS based animal tracking system," in *2008 International Conference on Intelligent Sensors, Sensor Networks and Information Processing*, 2008, pp. 617–622.

[7] [7] S. Kim, D. Kim, and H. Park, "Animal Situation Tracking Service Using RFID, GPS, and Sensors," in *2010 Second International Conference on Computer and Network Technology*, 2010, pp. 153–156.

[8] [8] L. Camacho, R. Baquerizo, J. Palomino, and M. Zarzosa, "Deployment of a Set of Camera Trap Networks for Wildlife Inventory in Western Amazon Rainforest," *IEEE Sens. J.*, vol. 17, no. 23, pp. 8000–8007, Dec. 2017.

[9] [9] R. Kays *et al.*, "Camera traps as sensor networks for monitoring animal communities," in *2009 IEEE 34th Conference on Local Computer Networks*, 2009, pp. 811–818.

[10] [10] W. Feng, J. Zhang, C. Hu, Y. Wang, Q. Xiang, and H. Yan, "A novel saliency detection method for wild animal monitoring images with WMSN," *J. Sensors*, vol. 2018, 2018.

[11] [11] F. N. Iandola, S. Han, M. W. Moskewicz, K. Ashraf, W. J. Dally, and K. Keutzer, "S QUEEZE N ET : A LEX N ET - LEVEL ACCURACY WITH 50 X FEWER PARAMETERS AND < 0 . 5MB MODEL SIZE," pp. 1–13, 2017.

[12] [12] K. Simonyan and A. Zisserman, "Very deep convolutional networks for large-scale image recognition," *3rd Int. Conf. Learn. Represent. ICLR 2015 - Conf. Track Proc.*, pp. 1–14, 2015.

[13] [13] K. He, X. Zhang, S. Ren, and J. Sun, "Deep residual learning for image recognition," *Proc. IEEE Comput. Soc. Conf. Comput. Vis. Pattern Recognit.*, vol. 2016-Decem, pp. 770–778, 2016.

[14] [14] 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, 2020.

[15] [15] 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.

[16] [16] A. G. Howard *et al.*, "MobileNets: Efficient Convolutional Neural Networks for Mobile Vision Applications," *CoRR*, vol. abs/1704.0, 2017.

[17] [17] W. Wang, Y. Li, T. Zou, X. Wang, J. You, and Y. Luo, "A Novel Image Classification Approach via Dense-MobileNet Models," *Mob. Inf. Syst.*, vol. 2020, p. 7602384, 2020.

[18] [18] W. Wang, Y. Hu, T. Zou, H. Liu, J. Wang, and X. Wang, "A New Image Classification Approach via Improved MobileNet Models with Local Receptive Field Expansion in Shallow Layers," *Comput. Intell. Neurosci.*, vol. 2020, p. 8817849, 2020.

[19] [19] B. Koonce and B. Koonce, "MobileNet v2," *Convolutional Neural Networks with Swift Tensorflow*, pp. 99–107, 2021.

[20] [20] A. Howard, W. Wang, G. Chu, L. Chen, B. Chen, and M. Tan, "Searching for MobileNetV3 Accuracy vs MADDs vs model size," *Int. Conf. Comput. Vis.*, pp. 1314–1324, 2019.

[21] [21] J. Sivasamy and T. S. Subashini, "Classification and predictions of Lung Diseases from Chest X-rays using MobileNet," vol. XII, no. 0886, pp. 665–672, 2020.

[22] [22] F. N. Iandola, M. W. Moskewicz, K. Ashraf, S. Han, W. Dally, and K. Keutzer, "SqueezeNet: AlexNet-level accuracy with 50x fewer parameters and <1MB model size," *ArXiv*, vol. abs/1602.0, 2017.

[23] [23] Y. Xu, G. Yang, J. Luo, and J. He, "An Electronic Component Recognition Algorithm Based on Deep Learning with a Faster SqueezeNet," *Math. Probl. Eng.*, vol. 2020, 2020.

[24] [24] H. Özcan, B. G. Emiroğlu, H. Sabuncuoğlu, S. Özdoğan, A. Soyer, and T. Saygı, "A comparative study for glioma classification using deep convolutional neural networks," *Math. Biosci. Eng.*, vol. 18, no. 2, pp. 1550–1572, 2021.

[25] [25] M. Hassanpour and H. Malek, "Document Image Classification using SqueezeNet Convolutional Neural Network," *5th Iran. Conf. Signal Process. Intell. Syst. ICSPIS 2019*, no. December, pp. 18–19, 2019.

[26] [26] K. N. Akpinar, S. Genc, and S. Karagol, "Chest X-Ray Abnormality Detection Based on SqueezeNet," *2nd Int. Conf. Electr. Commun. Comput. Eng. ICECCE 2020*, no. June, pp. 12–13, 2020.

[27] [27] K. Rantelobo, M. A. Indraswara, N. P. Sastra, D. M. Wiharta, H. F. J. Lami, and H. Z. Kotta, "Monitoring Systems for Counting People using Raspberry Pi 3," *2018 Int. Conf. Smart Green Technol. Electr. Inf. Syst. Smart Green Technol. Sustain. Living, ICSGTEIS 2018 - Proceeding*, vol. 7, pp. 57–60, 2018.

[28] [28] Lilygo, "TTGO T-Beam," *RIOT Documentation*, 2013. [Online]. Available: http://api.riot-os.org/group__boards__esp32__ttgo-t-beam.html. [Accessed: 31-Jan-2021].

[29] [29] J. Parham, J. Crall, C. Stewart, T. Berger-Wolf, and D. Rubenstein, "Animal population censusing at scale with citizen science and photographic identification," *AAAI Spring Symp. - Tech. Rep.*, vol. SS-17-01-, pp. 37–44, 2017.

[30] [30] LILA BC, "Snapshot Enonkishu," *Labeled Information Library of Alexandria: Biology and Conservation*, 2018. [Online]. Available: http://lila.science/datasets/snapshot-enonkishu. [Accessed: 12-Aug-2020].

[31] [31] LILA BC, "Snapshot Mountain Zebra," *Labeled Information Library of Alexandria: Biology and Conservation*. [Online]. Available: http://lila.science/datasets/snapshot-mountain-zebra. [Accessed: 12-Dec-2020].

[32] [32] LILA BC, "Snapshot Kruger," *Labeled Information Library of Alexandria: Biology and Conservation*. [Online]. Available: http://lila.science/datasets/snapshot-kruger. [Accessed: 12-Dec-2020].

[33] [33] 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.

[34] [34] LILA BC, "WCS Camera Traps," *Labeled Information Library of Alexandria: Biology and Conservation*. [Online]. Available: http://lila.science/datasets/wcscameratraps. [Accessed: 12-Dec-2020].

[35] [35] unsplash, "Llama," *Unsplash*. [Online]. Available: https://unsplash.com/s/photos/llama. [Accessed: 12-Dec-2020].

[36] [36] A. Dutta and A. Zisserman, "The {VGG} Image Annotator {(VIA)}," *CoRR*, vol. abs/1904.1, 2019.

[37] [37] W. Weng and X. Zhu, "INet: Convolutional Networks for Biomedical Image Segmentation," *IEEE Access*, vol. 9, pp. 16591–16603, 2021.

[38] [38] "Cayenne," *The Things Network*. [Online]. Available: https://www.thethingsnetwork.org/docs/applications/cayenne/index.html. [Accessed: 10-Jan-2020].