IJHSR
ijhighschoolresearch.org
■ RESEARCH ARTICLE

Poacher Activity Detection Device for Wildlife Conservation

Shreya Kakhandiki
Lynbrook High School, 1280 Johnson Ave, San Jose, CA 95129, USA; shreya.kakhandiki@gmail.com

ABSTRACT: The scale of the wilderness makes it extremely difficult to constantly patrol for poaching activity. Consequently, there is an urgent need for an effective automated tool to curb poaching. The goal of this project was to build a low-cost device that can be installed in wilderness areas as well as on surveillance drones and can send real-time alerts of poaching activity to park officials. The device hardware consists of a Raspberry Pi computer with a power bank, camera, and an antenna. The test data mainly consisted of publicly available images of wild animals and humans in the wilderness. An image classification algorithm was built using a set of pre-trained AI models to detect the presence of humans, weapons, and animals in a photo; this algorithm was implemented on a device that can be used to notify officials of possible poaching activity. The device successfully meets all engineering goals. The design costs about US$100, takes about 5 minutes to run per image, is over 90% accurate in recognizing humans, weapons, and animals in all the test images, and can generate notifications. The poacher activity detection device may help rangers monitor large areas and decrease poaching activity.

KEYWORDS: Ecology; Camera Trap; Poacher Detection; Wildlife Conservation; Artificial Intelligence; Machine Learning; Facial Recognition; Raspberry Pi.

■ Introduction

Wildlife poaching poses an ongoing, significant threat to global biodiversity.[1,2] Numerous studies have demonstrated that poaching is causing significant population declines in several more common species[3-5] and benefits insurgent groups and criminal networks around the world.[3]

Preventative action requires significant resources and constant monitoring. Moore *et al*.[8] looked at the effect of ranger posts and patrols in detecting and mitigating poaching threats, examining factors influencing the spatio-temporal dynamics of these threats, and testing the efficiency of management actions. This study highlights the need to increase patrol resources. Unfortunately, they require significant budget expansion which is often an intractable problem in developing countries. Furthermore, depending on the geographical region, the size and scale of reserve land makes it extremely hard for constant human patrolling. There is an inherent human safety issue as well; hundreds of rangers and volunteers have been killed by poachers.[9,10] As a result of this, poaching is often only recognized after the fact, at the risk of both animals and rangers. We have a clear and urgent need for an automated real-time tool to detect and curb poaching.

The goal of this project is to introduce a low-cost, scalable approach to wildlife conservation by developing a wildlife monitoring device that can alert the appropriate agency in real-time when poaching activity is detected. The entire system needs to be easy to install and maintain, widely deployable in any geographical region, offers one of the lowest cost solutions, and is based on standard components widely available today to promote a self-serve approach for anyone who wishes to deploy such devices in mass quantities. Releasing the device designs publicly will ensure that a maximum number of people will be able to build and deploy these devices rapidly and extensively for immediate impact on our planet's dwindling wildlife.

The project leverages recent innovations in computing hardware and software, motion sensors, AI-based image classification algorithms, solar cells, and satellite technology. This is packaged into a reference architecture design that anyone can reproduce, complete with hardware specifications, assembly instructions, and software components for rapid deployments anywhere in the world.

Related work:

Current preventative efforts and solutions to poaching can be classified into the following three main categories: risk prediction, wildlife monitoring, and threat monitoring.

Risk prediction:

Poaching risk prediction techniques have evolved from statistical analyses in the past to current approaches based on artificial intelligence (AI). The goal of these solutions is to predict where the poaching threat is greatest so appropriate resources can be mobilized in these areas. A potential shortcoming of this class of solutions is that poachers can simply move to other areas to avoid detection.

Risk prediction techniques can be further divided into two categories: statistical and GIS-based techniques, and AI-based techniques.

Statistical and GIS-based risk prediction

These techniques use statistical analyses incorporating geospatial elements of the physical environment to create risk maps indicating the highest areas of risk.[11] Typical high-risk areas occur near roads or water features that make it easy for poachers to haul animals to their destination.

AI-based risk prediction

These techniques use AI to predict when and where poachers are most likely to strike. Protection Assistant for Wildlife Security (PAWS)[12] is an AI system that predicts poaching

DOI: 10.36838/v4i5.17

risk levels in different areas of a wildlife preserve and helps rangers patrol more efficiently.[13] PAWS's machine-learning algorithm uses data from past patrols to predict where poaching is likely to occur in the future. It also uses a game-theory model to generate randomized, unpredictable patrol routes so poachers can't see any patterns to dodge rangers.

Wildlife Monitoring using image analysis:

Wildlife monitoring techniques consist of identifying animal populations and even individual animals. Such techniques help gauge the decline of specific species in a region. These techniques are effective when supported by wider regional coverage and a large number of images taken from several different vantage points. A great example of this is the Wildbook project,[14] which identifies individual animals in images uploaded by conservation scientists, rangers, and tourists.

Threat monitoring:

Monitoring of poaching threats via aerial surveillance

These techniques typically use manned planes or unmanned drones for aerial surveillance, taking pictures and videos that can be analyzed, often using computer-vision-based analysis to recognize animals and humans.[15] A potential drawback of aerial surveillance is that it requires ultra-high-resolution cameras (due to long distances to the subjects) that are very expensive. Photos from the above may miss capturing important features needed for species detection. Another downside to this technique is that it only works with large species in open habitats (i.e., savannas and whales in the ocean). Manned planes are expensive besides leaving a large carbon footprint impacting the very environment that wildlife relies on to survive. Consequently, these techniques are very difficult to scale worldwide.

Monitoring of poaching threats via camera traps

Camera traps are a relatively inexpensive method for capturing high-quality photos of animals and can be installed on a large scale for wide regional coverage. Due to these characteristics, camera traps are a useful tool in combating poaching. Examples of these include the Resolve organization, which, with Intel, has developed TrailGuard AI,[16] an anti-poaching camera trap system. While it has a long battery life and is capable of running AI models on the device, it still costs around $800 to manufacture. Maintenance costs are relatively high due to the lack of modular, replaceable components.

Methods

Device requirements:

The project goals can be translated into specific requirements for the poacher activity detection device prototype. The requirements include cost requirements (cost to assemble, cost to replace/repair), physical requirements (weight, form factor), performance requirements (accuracy, power consumption), to functional requirements (motion detection, animal and human detection, alerting authorities based on different conditions). These requirements were formulated such that the devices can be deployed practically with fairly high density in high-risk poaching areas.

Cost requirements:

The device should be low-cost, preferably costing less than US $100 to make it viable to deploy in arrays to cover large areas of wildlife preserves and parks. It should consist of standard, widely available components that can be individually replaced in case of malfunction or failure to reduce the cost of maintenance.

Physical requirements:

The device needs to weigh less than 500 grams to make it very portable and install it in hard-to-reach places and can be carried by small drones. It also needs to have a compact form factor (not bigger than 10cm x 10cm) that makes it easy to camouflage for discrete installations that can go undetected by poachers to prevent vandalism and theft.

Power requirements:

The device should run on natural renewable power sources such as solar so it can be installed in remote areas as well as surveillance drones that may not have access to regular power supplies. It should be efficient in power consumption to continuously run on small solar panels. It should also incorporate motion sensors and take pictures only when motion is detected, thereby reducing the volume of data to be transmitted over long distances. In habitats where daylight is limited, such as dense forests, the devices could run on battery power, and they would need to be maintained by park rangers.

Functional requirements:

The device should be able to run AI algorithms locally and should leverage widely available, commodity hardware.

Software on the device should be able to process images and detect animals, humans, or weapons, analyze these images for conditions that indicate a poaching threat, and notify appropriate authorities. To differentiate between rangers and possible threats, the device should be able to perform facial recognition on any humans in the frame. Based on the threat level it detects, the device should be able to send near-real-time alerts (within 10 minutes) to park rangers so they can respond quickly and effectively. It should be capable of detecting animals and humans with at least 90% accuracy to be an effective tool.

Device design:

We elected to use a Raspberry Pi with GSM / GPRS / GNSS modem (Figure 1) as the device hardware since it meets the requirements of being low-cost, small, low-power (10 watts), and sufficiently compute-capable (1.5 GHz, quad-core CPU) to run AI models. The Raspberry Pi is widely available at a cost of US $35. Other devices were considered but rejected either due to insufficient computing power to run AI models, higher costs, or insufficient connectivity options. The Raspberry Pi also has some key advantages in creating a device that is optimized for performance and functionality. It can run Linux, which has a good cadence of security and functionality updates that can be applied remotely. It also has a set of pins that can be easily connected to a wide variety of sensors and programmed. Storage is on standard SD cards which are available widely and cheaply. SD cards can be imaged from the internet for instant-on capability for the system. The Raspberry Pi can run Python development and

DOI: 10.36838/v4i5.17

runtime environments along with all the associated AI libraries and models that are necessary for training and automated detection.

Figure 1: Raspberry Pi board

After deciding to use a Raspberry Pi as the device's computer, a prototype was built with a small solar power bank, a camera, an infrared motion sensor, and a GPRS modem. This entire prototype setup (Figure 2) weighed less than 500 grams and can be reduced with further prototyping. The exposure of the motion sensor and the camera can also be reduced greatly as we fine-tune the prototype. Smaller camera form factors are currently being explored for this purpose. The size of the structure could be substantially reduced as well in future iterations.

Figure 2: Device set-up.

The next step was camouflaging the device (Figure 3). The prototype was placed in a structure that looks like a bird's nest. The entire device was hidden inside with only the motion sensor and the camera exposed, and the "bird's nest" was placed on a tree branch and further camouflaged with foliage from the tree. The overall system can be hidden inside a tree trunk or among rocks by covering an optional enclosure in some soil etc. Of all the aspects of the device, camouflaging is perhaps the easiest to improve on, as we can change the outer casing of the device to match any environment.

Figure 3: Camouflaged device.

Algorithm design:

The core algorithm of the device consists of various components working together based on certain trigger events and the contents of the photos taken by the camera. Details of the overall workflow combining the algorithms can be found in Figure 4 below. A high-level description of the algorithm is as follows:

1. A photo is taken by the camera when the motion sensor is triggered due to motion in the environment.
2. The photo is analyzed for the presence of animals and/or humans as well as for the presence of any weapons.
3. Further analysis is done to classify the animal species and check whether the human is a local park ranger or an unknown individual.
4. The appropriate alert level is set triggering a potential alert to park/preserve authorities, e.g., if an unknown human with a weapon is detected, this triggers the highest poaching threat level (see detailed description of alert logic in a later section).

Software components:

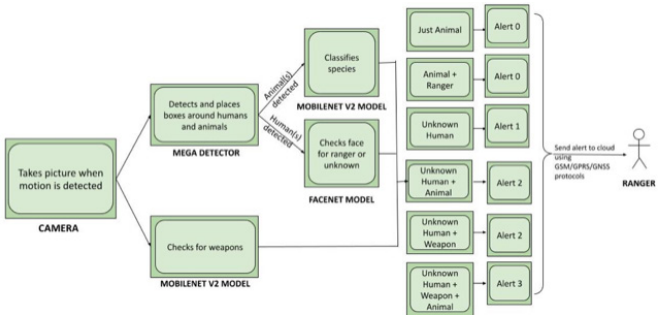

Figure 4: Device set-up.

Once this approach was decided on, the appropriate AI libraries needed to be pinpointed for each of the four components.

Human/animal/vehicle detection:

The MegaDetector[17] model was found to be best suited for detecting humans, animals, and vehicles in images. The MegaDetector model is specially trained for conservation biologists to identify humans and animals in camera trap images from a variety of ecosystems. MegaDetector is used in the current system to avoid running species classification and ranger identification on empty images, as well as to crop the relevant objects from the image to speed up downstream analysis (Figure 5).

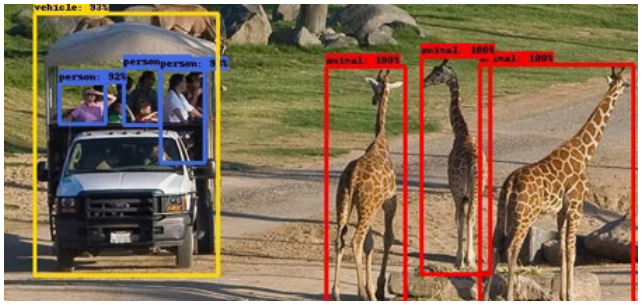

Figure 5: Detection of animals, humans, and vehicles with *MegaDetector*.

Weapon detection and species classification:

MobileNetV2 is a lightweight model pre-trained on weapons and animal species from the ImageNet database and is highly suited for our device here. Given the constraints of the high speed and low memory of the Raspberry Pi device, we selected the MobileNet v2[18] classifier based on an analysis done by Bianco *et al.*[19] Some of the other networks that were evaluated did not have versions compatible with the Raspberry Pi and were hence ruled out.

MobileNetV2 also serves the purpose of classifying animal species (Figure 6). It has very good accuracy for detecting a variety of animal species and new species are being added on an ongoing basis by researchers worldwide. At this time, MobileNetV2 was chosen for this purpose as it is widely available and is actively being enhanced by a large number of stakeholders.

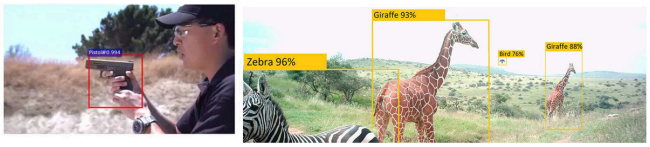

Figure 6: Detection of weapons and classification of species with MobileNetV2

Poacher/non-poacher classification:

To detect whether the human in the photo is a ranger or a potential poacher, we must perform face recognition. Identifying individual poachers is difficult since there is no readily available database of poachers. Even if such a database existed, it would become obsolete rapidly since one would need to add new poachers promptly.

To overcome these real-world constraints, we decided to train a face recognition system for known rangers, and mark everyone else as unknown. The database of rangers and other associated park personnel is much more reliable and can be kept up to date easily. The device identifies humans that are not poachers (rangers and other park officials). An alert is generated if the human is not recognized by the device, indicating a possible poacher. This approach can be further augmented in the future by adding photos of known poachers who pose a high threat to alert police authorities quickly in case such poachers are found by the device in a wildlife preserve.

For facial detection, the well-known FaceNet[20,21] model (Figure 7) was selected. FaceNet uses MTCNN with triplet loss for improved accuracy over other existing models. It is lightweight and only needs 2-3 images/person to train it to recognize the person. Furthermore, it is compatible with Raspberry Pi, making it highly suited for the poaching detection device.

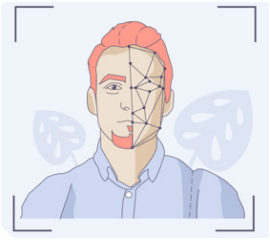

Figure 7: Detection of weapons and classification of species with MobileNetV2

Alert rules:

An important part of the overall design of the poaching detection device was designing the rules that would establish the threat levels and send out appropriate alerts (Figure 8). The logic for the alerting rules is based on the threat level perceived by the device per its analysis of the photo taken. It is as follows:

- If the photo contains only an animal, the threat level is 0, no alert needs to be sent.
- If the photo contains a human that is recognized as a ranger, the threat level is still 0, no alert needs to be sent.
- If the photo contains only a human that is unknown, the threat level is 1 and an alert needs to be sent that an unknown person is on the park / preserve land.
- If the photo contains an animal and an unknown human, the threat level is 2 and an alert needs to be sent that an unknown person is close to an animal and represents a poaching threat.
- If the photo contains an unknown human with a weapon, the threat level is 2 and an alert needs to be sent that an unknown person with a weapon is on the park / preserve land and represents a hunting / poaching threat.
- If the photo contains an animal with an unknown human with a weapon, the threat level is 3. This is the highest threat level, and an alert needs to be sent immediately that there is an imminent poaching threat.

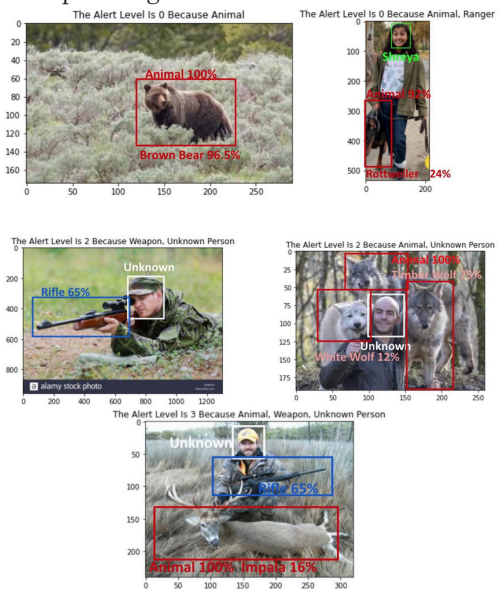

Figure 8: Examples of alert levels.

Results and Discussion

Each of the three components (MegaDetector, MobileNetV2, FaceNet) of the Protego device were separately tested for performance accuracy. The entire device was then tested for overall end-to-end performance. Performance metrics were chosen as appropriate for each component and are described in each section below.

Animal and human detection:

The MegaDetector network has been developed to detect three classes: animals, humans, and vehicles. It is used in our device for detecting the first two classes. Each class detection

DOI: 10.36838/v4i5.17

was tested separately, and three metrics were calculated for each:

- ROC-AUC (Receiver Operating Characteristics – Area Under the Curve), is created by plotting the true positive rate (TPR) against the false positive rate (FPR) at various threshold settings. The AUC measures the degree of separability between the classes being distinguished.
- Sensitivity or true positive rate refers to the proportion of positives correctly identified by the detector to the actual number of positives. It is a measure of how well a classifier can identify true positives.
- Specificity or true negative rate refers to the proportion of negatives correctly identified by the detector to the actual number of negatives. It is a measure of how well a detector can identify true negatives.

Testing was done on 204 negative images (no animals or humans in image), 58 images containing animals, and 35 images containing humans. These images were obtained from open-source images (including day/night captures from various angles) on the internet and were manually labeled. As seen in Figure 9, the ROC curves for both the animal and human classes are near perfect, with corresponding area under the curve (AUC), sensitivity, and specificity values shown below the plot.

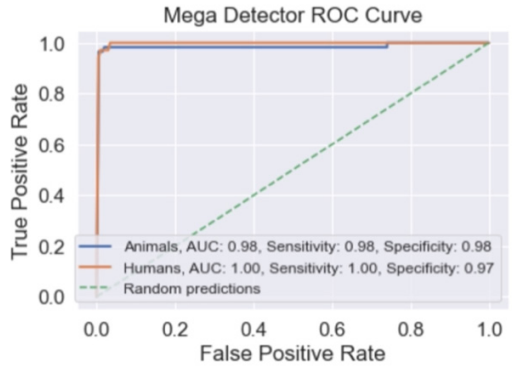

Figure 9: Mega Detector Test Results.

Weapon detection:

MobileNetV2 has been thoroughly tested for weapons detection accuracy by Mohebban[22] in the context of improving surveillance systems to address mass shootings. He tested the MobileNet detector on 152 images with no weapons, 152 images containing handguns, and 137 images containing rifles. Figure 10 shows the ROC curves plotted for all three classes. The AUC shows as greater than 90% for all classes indicating very good performance in detecting common weapons.

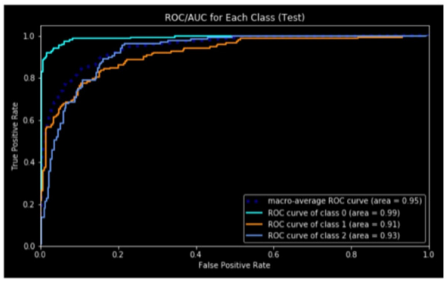

Figure 10: Weapons Detection Test Results.

Ranger identification:

The FaceNet code was tested on 25 "ranger" images (facial images of the author). When tested on a single image at a time, the FaceNet algorithm shows relatively poor accuracy (percent of images correctly identified). Hence a temporal filter was added which ran FaceNet on multiple consecutive images and the maximum detection value over those images is chosen as the final value. For example, if FaceNet identifies two out of three images as Ranger A, then the final output is set to Ranger A. Applying such a temporal filter significantly improves the accuracy of the algorithm as shown in Figure 11. Temporal filtering also improves robustness when there is target motion. An accuracy of over 90% was observed with multiple camera images. Device setting ensures successive capture of 5+ images to get to the highest accuracy.

Test Accuracy		
Single	Max of 3	Max of 5
72%	92%	96%

Figure 11: FaceNet test results.

Results

Finally, we measure the end-to-end performance of the entire device. Since the Protego device is built to classify the input image into four classes (alert levels), we use the confusion matrix as a metric to measure its performance.

A confusion matrix is a summary of prediction results on a classification problem. The number of correct and incorrect predictions are summarized with count values and broken down by each class. This is the key to the confusion matrix. The confusion matrix shows how a classification model can be confused when it makes predictions. It gives one insight not only into the errors being made by your classifier but more importantly the types of errors that are being made.

Figure 12 shows the confusion matrix generated for the Protego device. A set of 40 images were used to test the device, evenly divided among the four classes or alert levels (10 images per class). As seen in the figure, the device performed extremely well on the test images. Average accuracy came in at 92.5%. The most reassuring aspect of the results is that in cases where the device failed to correctly generate the alert level, it was only off by one level, thus not significantly impacting the outcome.

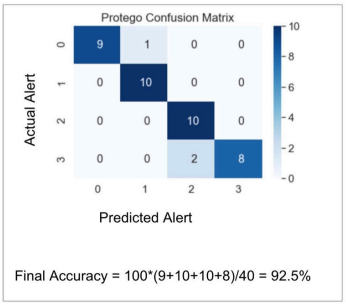

Figure 12: Overall device test results (92.5% accuracy).

Conclusion and future work

The proof of concept was successful: all design requirements were met, and accuracy was high under test conditions. This work sets up the foundation for a low-cost, open platform that can be greatly expanded upon by others. There is tremendous scope for improvement in the areas of performance and accuracy, which can be achieved by innovation and field testing of the device. We hope to build on and improve this device in the future.

For example, one of the future goals is to expand the algorithm to detect snares and other poaching tools so that rangers can remove them before any animals are caught. In addition to that, a night vision camera would help catch nighttime and low light poaching activities. While this will be more expensive than just an ordinary webcam, it will catch a much higher percentage of poachers. Other ideas include estimation of the proximity of poacher to animal and include information in the alert sent and connecting to the systems like Wildbook to effectively detect and identify individual animals to protect.

From the field deployment perspective, work needs to be done in deploying the device to high poaching risk areas and partnering with wildlife protection organizations to test the device.

Acknowledgements

I would like to thank Mr. Lester Leung, my Chemistry teacher and the school science fair coordinator for giving me the opportunity to conduct this research and represent my school at the Synopsis Science Fair. I would like to thank Prof. Tanya Berger-Wolf at the Ohio State University. She is the director and co-founder of the non-profit organization Wild Me, which utilizes computer vision for wildlife conservation. Dr. Berger-Wolf gave me the opportunity to work on a Wild Me project, which led to my inspiration to do this research. I would like to thank Prof. Charles Stewart at RPI for guiding me with my hypothesis and overall design goals.

References

1. The Devastating Effects of Wildlife Poaching - One Green Planet https://www.onegreenplanet.org/animalsandnature/the-devastating-effects-of-wildlife-poaching/ (accessed Nov 6, 2021)
2. 52 Eye-Opening Poaching Statistics You Must Know in 2021 https://petpedia.co/poaching-statistics/ (accessed Nov 6, 2021)
3. Anderson, B.; Jooste, J. Wildlife Poaching: *Africa's Surging Trafficking Threat, Africa Security Brief*, No. 28, May 2014
4. Everatt, K. T.; Kokes, R.; Lopez Pereira, C. Evidence of a Further Emerging Threat to Lion Conservation; Targeted Poaching for Body Parts. *Biodivers. Conserv.* **2019**, *28*, 4099–4114
5. Linkie, M.; Martyr, D. J.; Holden, J.; Yanuar, A.; Hartana, A. T.; Sugardjito, J.; Leader-Williams, N. Habitat Destruction and Poaching Threaten the Sumatran Tiger in Kerinci Seblat National Park, Sumatra. *Oryx* **2003**, *37*
6. De Sadeleer, N.; Godfroid, J. The Story behind COVID-19: Animal Diseases at the Crossroads of Wildlife, Livestock and Human Health. *Eur. j. risk regul.* **2020**, *11*, 210–227
7. Karesh, W. B.; Cook, R. A.; Bennett, E. L.; Newcomb, J. Wildlife Trade and Global Disease Emergence. *Emerging Infect. Dis.* **2005**, *11*, 1000–1002
8. Moore, J. F.; Mulindahabi, F.; Masozera, M. K.; Nichols, J. D.; Hines, J. E.; Turikunkiko, E.; Oli, M. K. Are Ranger Patrols Effective in Reducing Poaching-Related Threats within Protected
9. Over One Thousand Park Rangers Die in 10 Years Protecting Our Parks and Wildlife | News | Global Conservation https://globalconservation.org/news/over-one-thousand-park-rangers-die-10-years-protecting-our-parks/ (accessed Nov 6, 2021)
10. O'Grady, C, The Price Of Protecting Rhinos: Conservation has become a war, and park rangers and poachers are the soldiers, *The Atlantic*, **2020**
11. Shaffer, M. J.; Bishop, J. A. Predicting and Preventing Elephant Poaching Incidents through Statistical Analysis, GIS-Based Risk Analysis, and Aerial Surveillance Flight Path Modeling. *Tropical Conservation Science* **2016**, 9, 525–548
12. Outsmarting poachers https://www.seas.harvard.edu/news/2019/10/outsmarting-poachers (accessed Nov 6, 2021)
13. Fang, F.; Nguyen, T. H.; Pickles, R.; Lam, W. Y.; Clements, G. R.; An, B.; Singh, A.; Schwedock, B. C.; Tambe, M.; Lemieux, A. PAWS — A Deployed Game-Theoretic Application to Combat Poaching. *AIMag* **2017**, 38, 23–36
14. Home | Wild Me https://www.wildme.org/#/ (accessed Nov 6, 2021)
15. Chalmers, C.; Fergus, P.; Wich, S.; Montanez, A. C. Conservation AI: Live Stream Analysis for the Detection of Endangered Species Using Convolutional Neural Networks and Drone Technology. *arXiv* **2019**
16. Trailguard - RESOLVE https://www.resolve.ngo/trailguard.htm (accessed Nov 6, 2021)
17. CameraTraps/megadetector.md at master · microsoft/CameraTraps · GitHub https://github.com/microsoft/CameraTraps/blob/master/megadetector.md (accessed Nov 6, 2021)
18. Sandler, M.; Howard, A.; Zhu, M.; Zhmoginov, A.; Chen, L.-C. Mobilenetv2: Inverted Residuals and Linear Bottlenecks. *In 2018 IEEE/CVF Conference on Computer Vision and Pattern Recognition*; IEEE, 2018; pp. 4510–4520
19. Bianco, S.; Cadene, R.; Celona, L.; Napoletano, P. Benchmark Analysis of Representative Deep Neural Network Architectures. *IEEE Access* **2018**, 6, 64270–64277
20. Face Recognition with FaceNet and MTCNN – Ars Futura https://arsfutura.com/magazine/face-recognition-with-facenet-and-mtcnn/ (accessed Nov 6, 2021)
21. GitHub - R4j4n/Face-recognition-Using-Facenet-On-Tensorflow-2.X https://github.com/R4j4n/Face-recognition-Using-Facenet-On-Tensorflow-2.X (accessed Nov 6, 2021)
22. GitHub - HeeebsInc/WeaponDetection https://github.com/HeeebsInc/WeaponDetection (accessed Nov 6, 2021)

Author

Shreya Kakhandiki is a senior at Lynbrook High School in San Jose, CA. She is passionate about wildlife conservation, computer science, and public policy. She is fascinated by how these fields intersect and would like to pursue a career in wildlife conservation advocacy.

DOI: 10.36838/v4i5.17