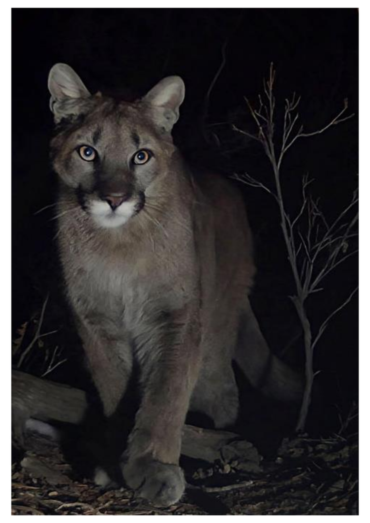


PumaGuard: AI-enabled Targeted Puma Mitigation

Aditya Viswanathan, Adis Bock, Zoe Bent. Tate Plohr, Suchir Jha, Celia Pesiri, Pheobe Reid

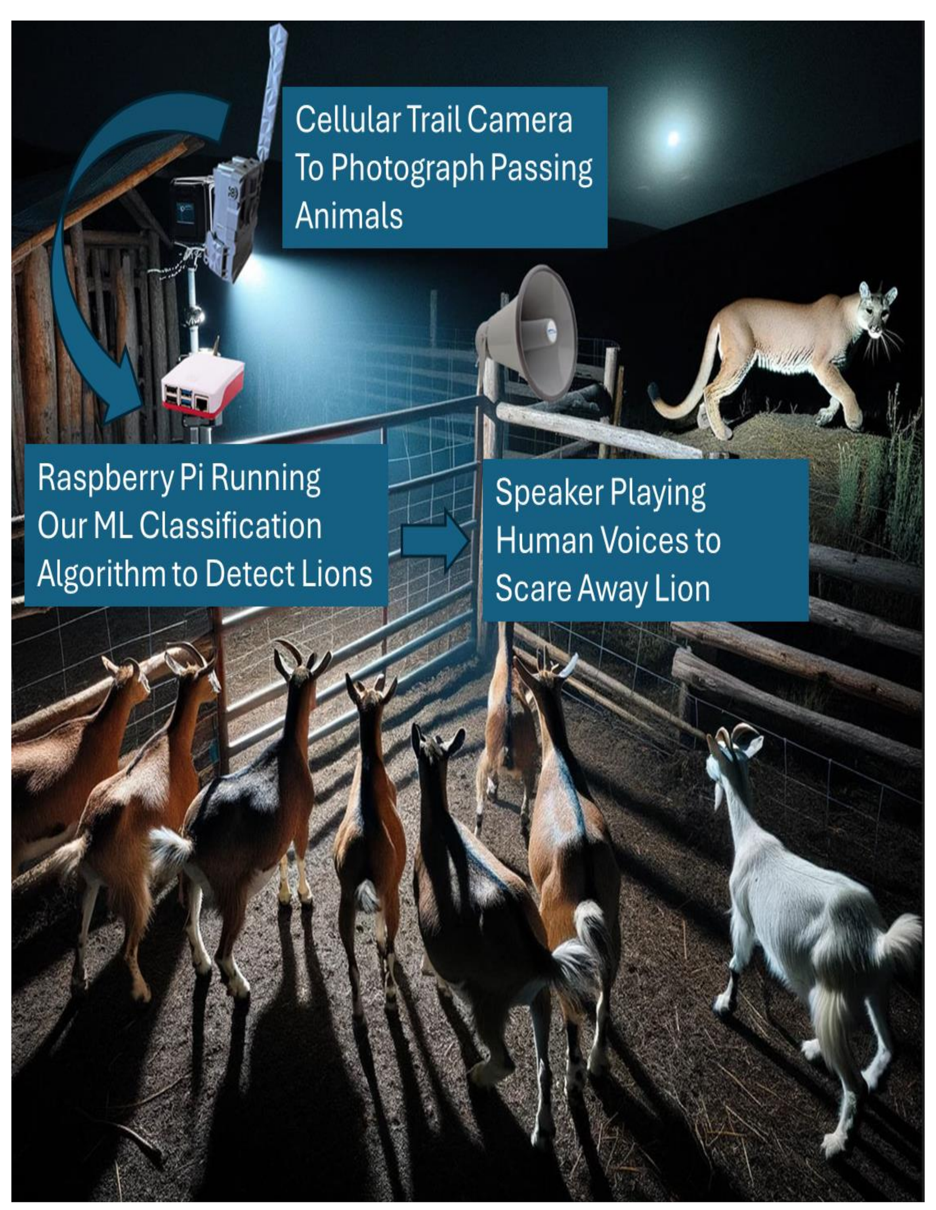
The goal of our project is to reduce conflicts between pumas and human interests

- Habitat loss due to human encroachment, wildfires, and drought
- This has resulted in increased human-wildlife conflict with pumas killing livestock at the local stables
- As a result of pumas killing livestock at the stables, 1-2 pumas per year have been killed by New Mexico Game and Fish in Los Alamos
- Catastrophic for both the puma population, overall ecosystem and for livestock owners with no end to this cycle in sight
- The stables are a high traffic area, and over time the pumas get habituated to constant sounds and lights, requiring a targeted solution beyond simple motion activated devices
- Puma attacks on livestock are a pressing issue as incidents have taken place as recent as October 2024 with subadults replacing previously killed pumas

Map of Los Alamos and surrounding National Parks where training data was collected

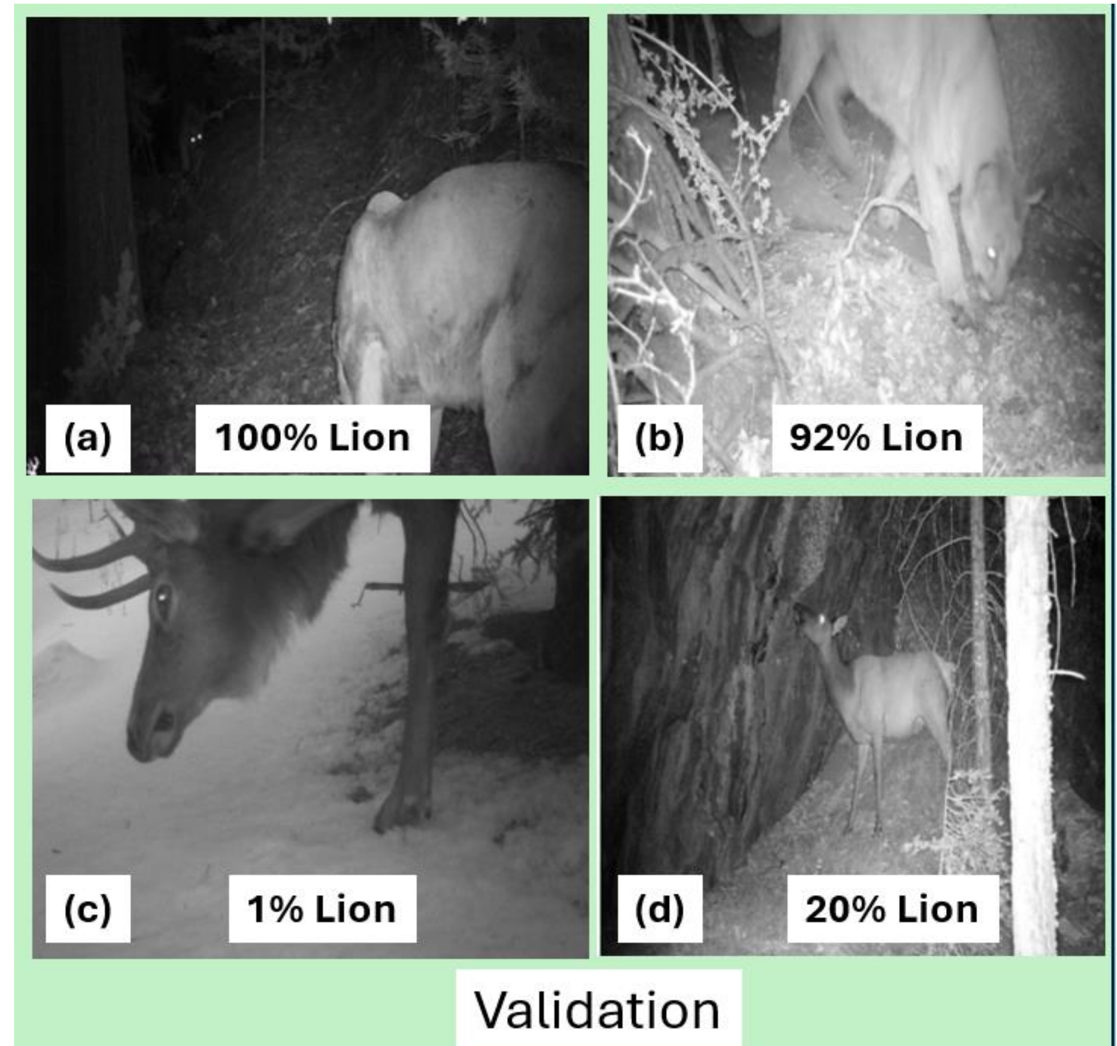
Map of Los Alamos stables with triangles indicating puma attacks in October 2024.

Our Proposed Solution

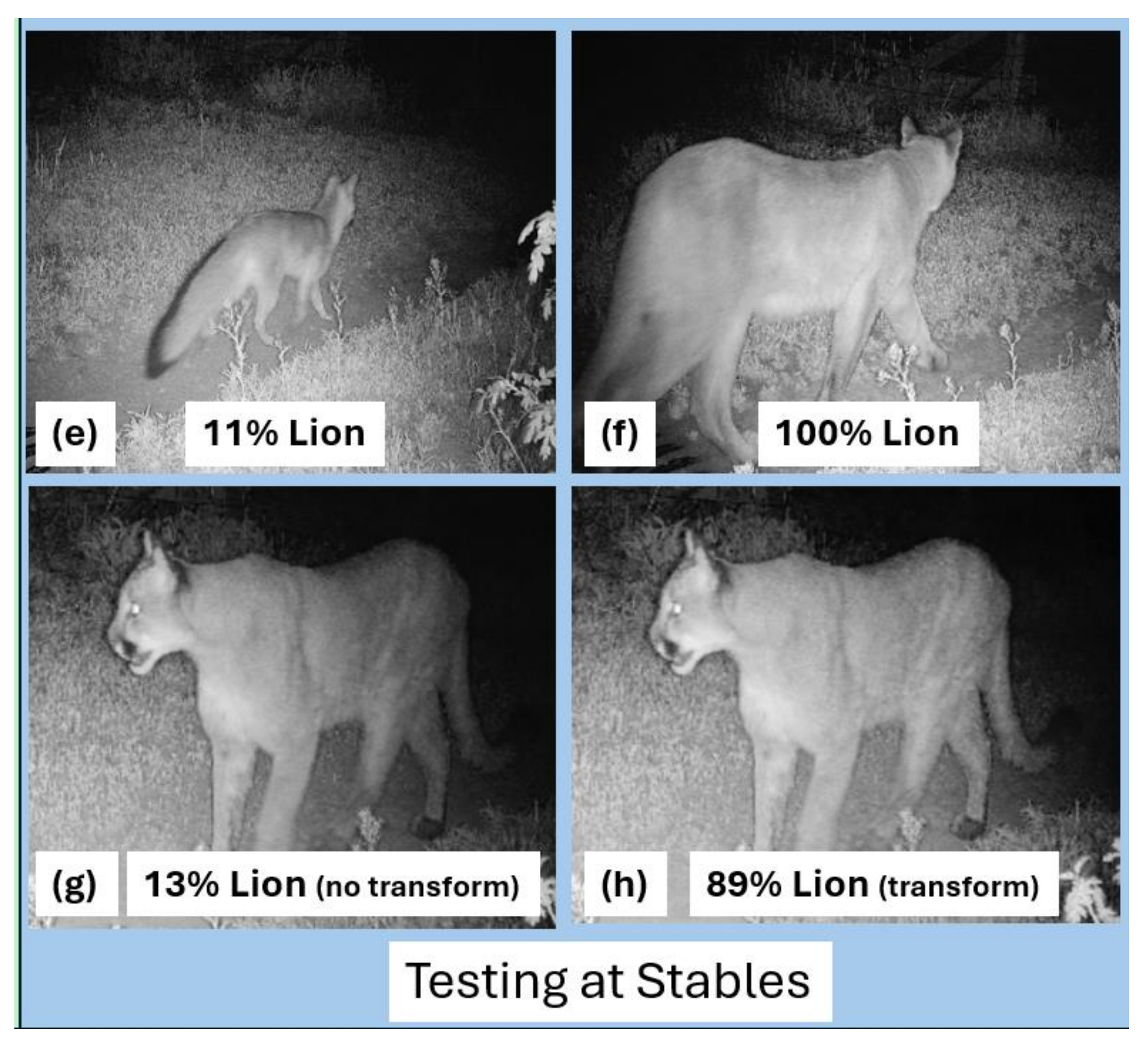


- Our targeted approach plays talk radio when a trail camera image is classified as a puma
- Image classification is built off of the CNN Xception model.
- Model trained on 1300+ trail cam images of pumas and other wildlife
- Our best model achieved 99% training accuracy and 91% validation accuracy
- **Over a set of 3 consecutive trail camera images per animal visit, the algorithm achieves 100% positive classification**
- False positives are rare (2%) and of low consequence
- Economical, flexible and scalable approach that can be tailored to specific scenario (e.g. different lights, sounds and sprinklers in random order can be deployed)

Results



Our dataset (collected at Bandelier National Monument and Valles Caldera National Preserve) was split into 80% for training and 20% for validation. We obtained 91% validation accuracy on our final algorithm. Additionally, we received no false negatives and 2% false positives. Above are some of our validation photos (anything above 50% is a lion).



We classified photos from the stables using our model with a high success rate. In the bottom left image (g), the algorithm failed at identifying the lion. This is because the lion is at the edge of the frame and not properly lit up. Increasing the brightness (h) resulted in the algorithm detecting the lion.

Results Continued

Table 1: Classification Algorithms. All models were run on a Google Cloud TPUv2. The Xception model was pre-trained on the IMAGENET [6] The models, weights, Python notebooks, and data are publically available at `https://extreme-lion-challenge.readthedocs.io/` [9].

	Case	Model	Epoch	No. Pics	Resolution	Accuracy Train	Accuracy Val.	Loss Train	Loss Val.	Time [s]
Poor convergence for all models with no pretraining	1	light	500	847	128 x 128	57%	69%	0.65	0.68	2762
	2	light	500	847	256 x 256	62%	75%	0.62	0.66	6683
	3	aug-light	600	847	256 x 256	51%	67%	0.67	0.66	7122
Xception without our training data did not work	4a 4b	pre-train	1 300	200	128 x 128	57% 100%	50% 80%	3.21 0.03	0.50 1.28	0 624
	5	pre-train	300	847	128 x 128	100%	88%	0.03	0.60	898
Our best model	6	pre-train	300	1302	512 x 512	99%	91%	0.08	0.32	1753

* Note Xception without weights performs poorly so pretraining was key to success

- Initial light algorithm showed poor results, and the validation accuracy was higher than the training accuracy (case 1). This didn't improve with increased resolution (case 2) or augmentations (case 3)
- The pretrained Xception model without our data performed poorly (case 4a)
- Combining the Xception model with our data worked well. Using 200 images had 80% validation accuracy (case 4b).
- **A pretrained model was critical to our success**
- Adding images increased accuracy (cases 5-6), with our best model reaching 91% validation accuracy.
- Since our camera setup takes 3 images every time the motion sensor is activated, 91% accuracy will result in no false negatives.

Remaining Challenges and Future Work

- Slow or nonexistent network connectivity on site: run our model near the camera for real-time deterrence (edge computing)
- May have no power on site: use low-power Raspberry Pi (no GPU)
- Commercial trail cam has inadequate control and slow transfer: use Pi camera
- We are working to deploy our machine learning algorithm onto a Raspberry Pi. attached to a camera and motion sensor
- We are also working on a lighter model, model quantization, and using NPU's for faster classification
- This AI-enabled solution is easy to implement, scales well to multiple cameras, and is inexpensive as opposed to building high fences or enclosed pens
- The workflow proposed may have broad utility with movements to reintroduce other predators to their previous environment