Wildlife Conservation using Unsupervised Clustering

Jordan Freyman

Challenges of Wildlife Monitoring

- During the height of species endangerment, wildlife monitoring is an impactful method of using our advancements in technology to help mitigate it.
- Manually scouring through countless footage and other captured data of animal habitats when checking for population trends is labor-intensive, prone to errors, and generally an inefficient use of time.
- Inefficient monitoring can lead to loss of critical information and misleading data, overall making the process of the data collection pointless.
- How can this be solved?

For Context…

- We are currently undergoing a biodiversity crisis, and wildlife conservation is more urgent than it ever has been.
- According to the World Wildlife Fund (WWF), global wildlife populations have declined by 69% on average since 1970. The International Union for Conservation of Nature (IUCN) lists over 42,000 species as threatened with extinction.
- Some statistics:
  - Only ~45,000 elephants left out in the wild
  - Zebras are also close to endangerment, with about 660,000 – 1 mil wild population
  - Several gazelle species are critically endangered, such as the dama gazelle (fewer than 500)

Purpose & Motivation

- With habitat loss, climate change, and poaching at an all-time high, using camera traps and advanced image analysis with automation could significantly help monitor population trends.
- Making efforts towards wildlife conservation using computational tools is crucial during such a dangerous time.
- One way to approach the issue of image analysis is through unsupervised clustering.

Discussing the Research Question

- **How can unsupervised clustering help analyze wildlife distribution patterns in camera trap images?**
- Or, more specifically, **how can image clustering techniques enhance wildlife recognition and population monitoring in wildlife datasets, and what factors affect clustering accuracy and efficiency?**
- Other considerations:
  - What models would perform best with clustering?
  - Why unsupervised learning over supervised?
  - How could results vary between usage of different data sets?
- Let’s discuss methodology.

I used THREE different image classification models to recognize animals in the Serengeti National Park, conducting unsupervised learning.

- Including:
    - VGG16, a pre-trained CNN model for image recognition with 16 layers
    - ResNet-50, a similar model with 50 layers
    - EfficientNetB0, using **237** layers!

- What results have I accomplished from using these models? And why use *unsupervised* learning?

The Benefits of Using Unsupervised Learning

- Data sets are often unlabeled. And labeling data is expensive and time-consuming!
- To achieve a goal using an unlabeled data set, unsupervised clustering can be a lot more useful since it groups similar images together without need for labels.
- And since there’s no need for labeled guidance with unsupervised clustering, a lot of time and resources are spared. This makes the method a lot more scalable.
- Not to mention the potential we get for discovering new patterns when we cluster without explicit guidance. This could lead to insights about population trends, animal behaviors, and habitats.
- Unsupervised clustering can also facilitate the labeling process by grouping visually similar images together, which can help with model training.

...And the Challenges

- Of course, unsupervised clustering does lead to more complications with interpreting data when nothing is labeled.
- Sometimes using a pre-trained model without any fine-tuning can result in poor clustering when the data set is too vague.
- When data sets contain less refined data, such as blurred or dim images, there will be more noise and inaccuracies in the resulting clusters.
- That being said, unsupervised clustering can be inefficient at times but proves to be an excellent tool when used in the right context with appropriate expectations.

The Data Set

- After many, many attempts, I settled for specialization in object detection, that being for recognizing animals within a data set of MILLIONS of images, where **72%** didn’t contain any animals at all!
- I’m talking about the Snapshot Serengeti Project — a data set that contains hundreds of thousands of images potentially containing wildlife, in a variety of camera angles and environmental changes.
- And how did the process go?

The Process

- First, I preprocessed a sample of images.
- As mentioned before, the entire data set contains approximately **7.1 million** images. With my very limited resources, trying to sit and use three separate models to search for animals in that many images would've been impossible.
- So, I cut down on the size by testing my script with only **12,030** images instead. Using the Google Cloud SDK Shell and searching through the labeling of the data set, I made sure to gather a decently diverse set of images for the models to sift through. I kept the images alone and disregarded the labels to properly conduct unsupervised learning.

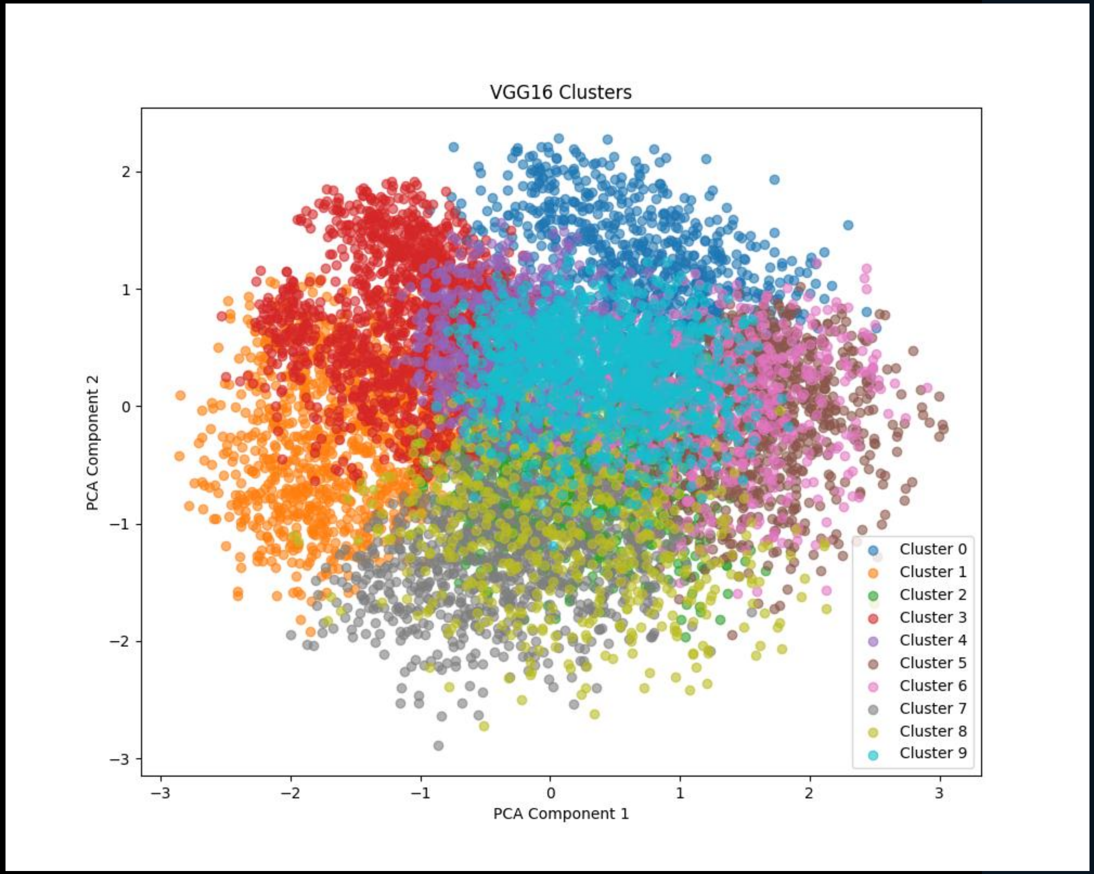
So, what are these models actually doing?

- All three models follow the process of extracting features from each image, encoding high-level information about texture, shape, patterns, and edges.
- The extracted features are reduced in complexity to make clustering easier, which is done with Principal Component Analysis.
- Then the program uses K-Means clustering to group images together based on their similarities.
- By detecting similarities in a large data set of images, the models can easier find images that include animals in a variety of different environmental conditions!

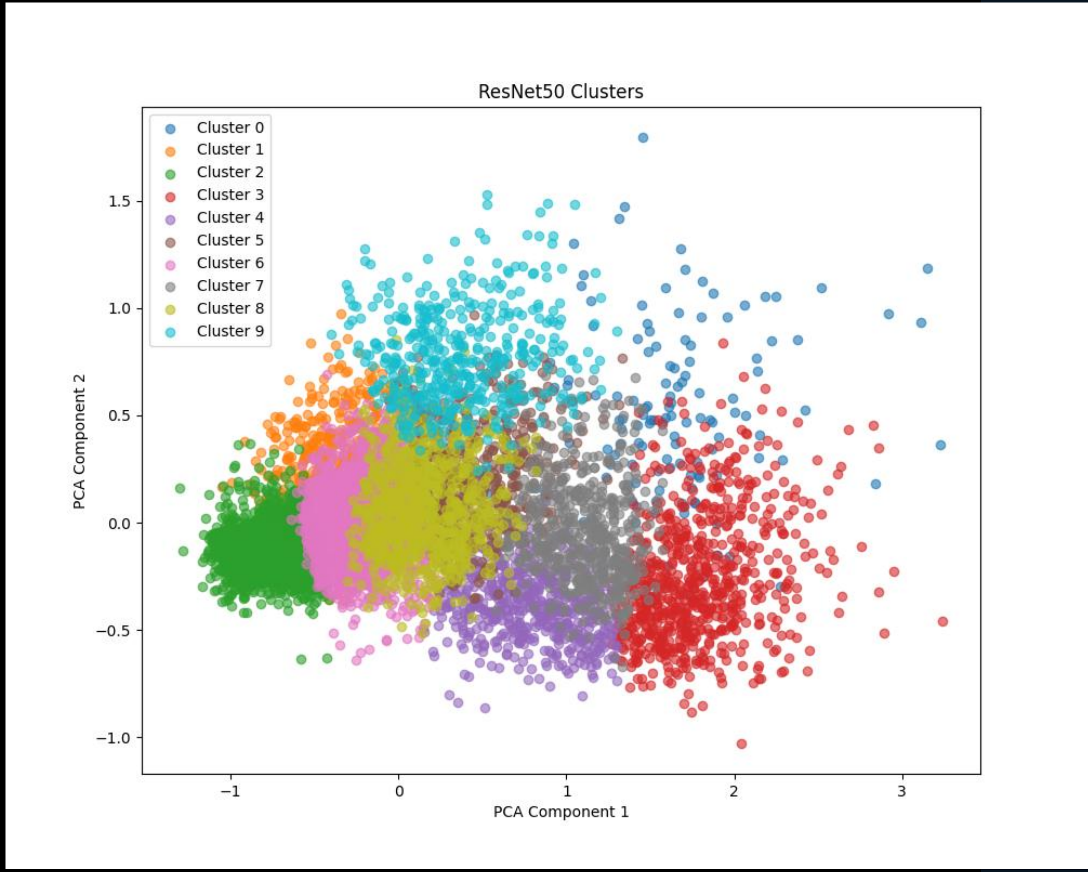
The Process cont'd

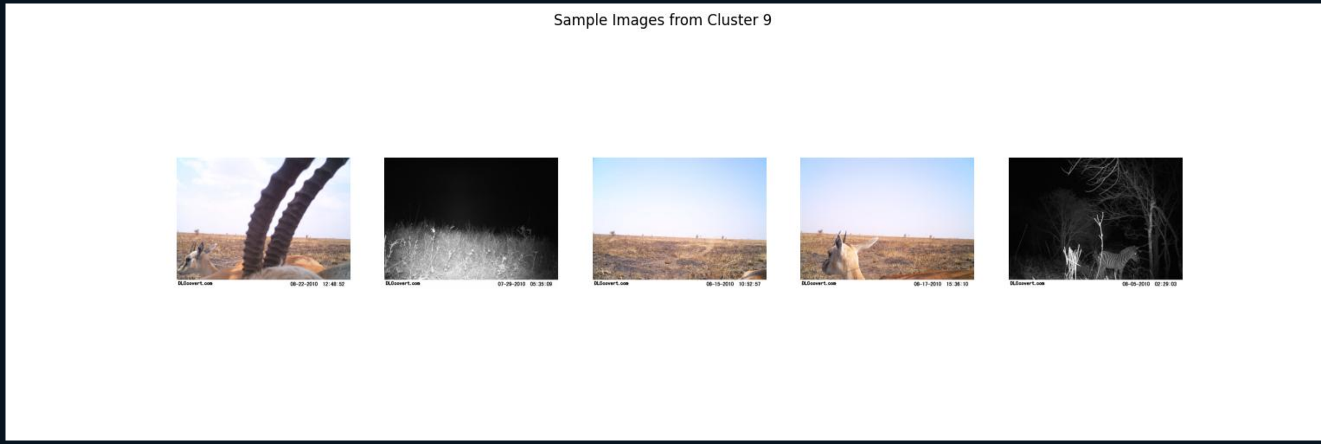
- Preprocessing images particularly means resizing them to 128x128 pixels, then normalizing the pixel values for more efficient model performance.
- As mentioned before, each model then processes the main features standing out from all 12,030 images, then applying PCA to reduce features to 50 principal components for clustering (in which the reduced features are cached for faster reuse.) Then comes clustering, followed by the most important part, the visualization. Three scatterplots are generated showing the clusters that all three models generated, as well as five sample images for every cluster generated.
- And so what were the results?

Sample images from VGG16 clusters

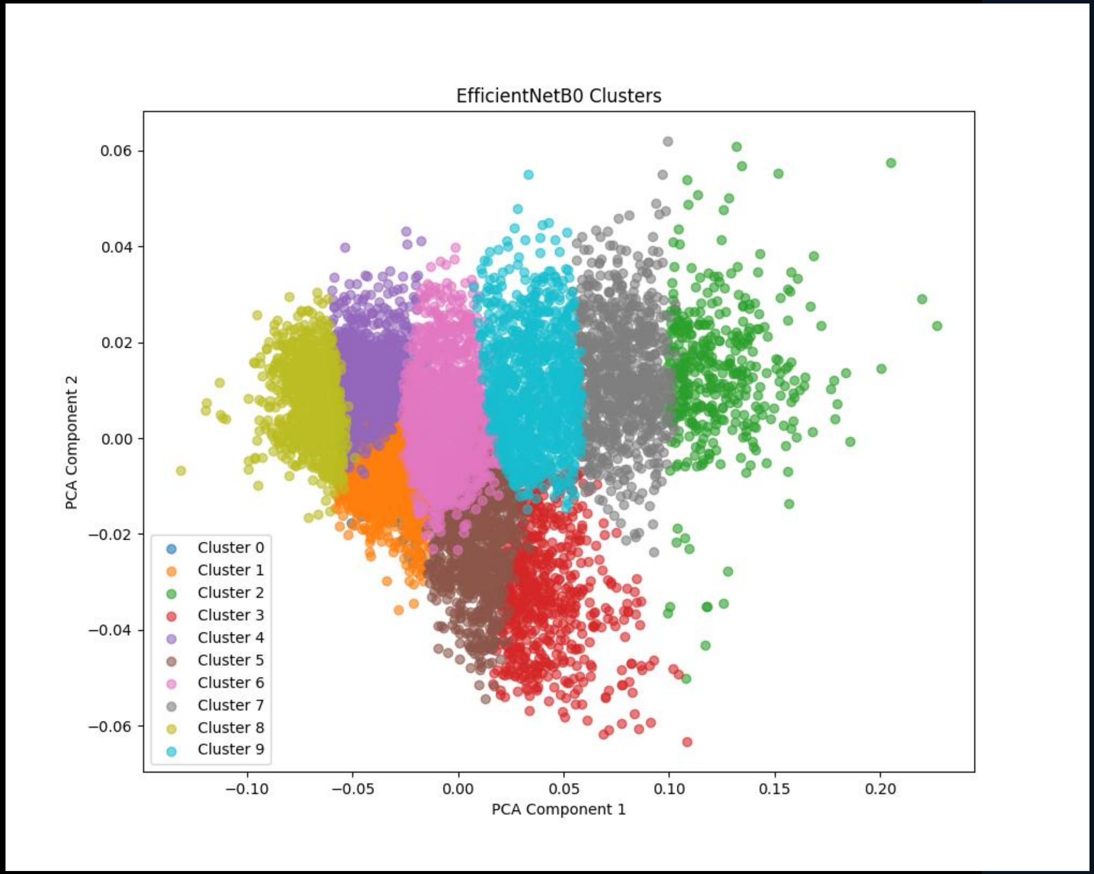


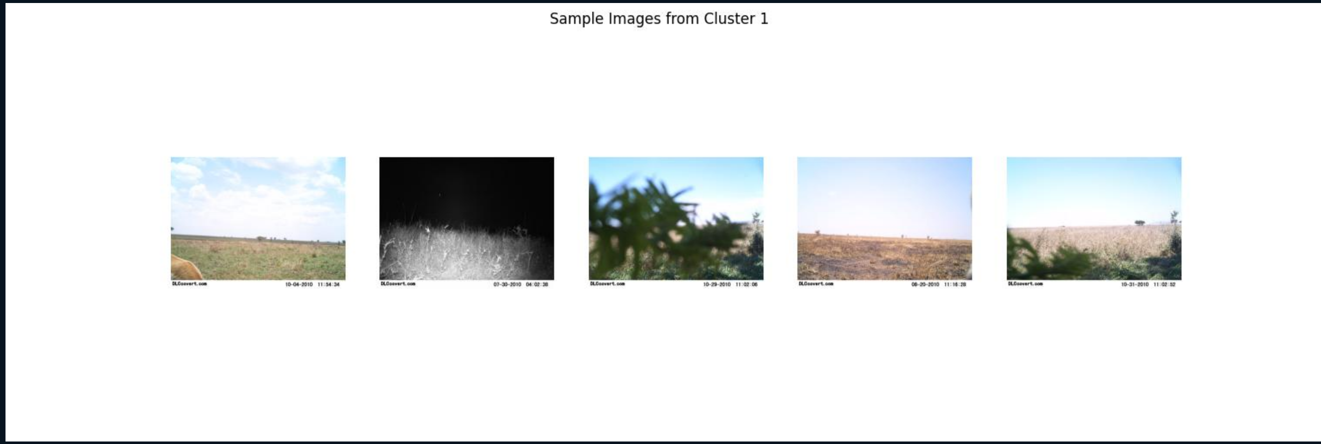
Sample images from ResNet50 clusters



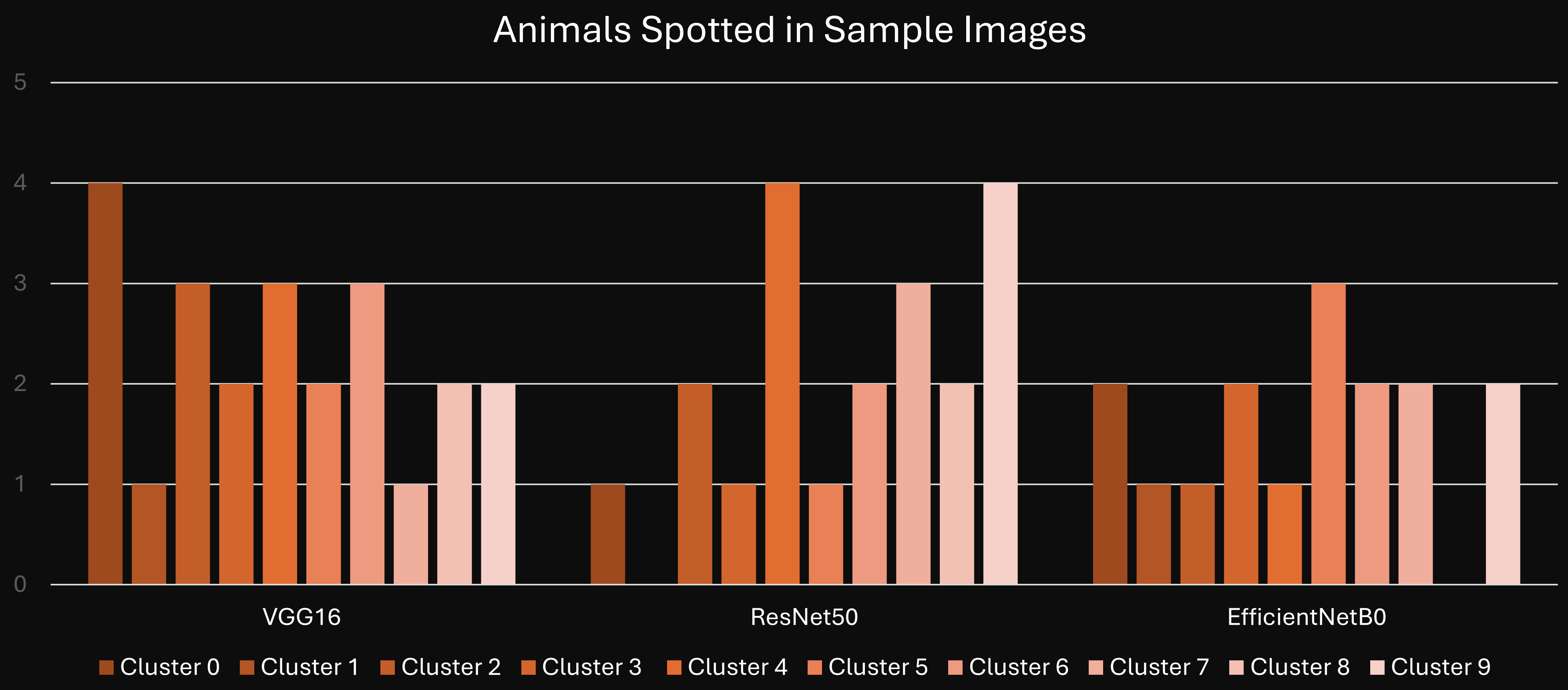


Sample images from EfficientNetB0 clusters





Results



Animals Spotted in Sample Images
5
4
3
2
1
0
VGG16
ResNet50
EfficientNetB0
Cluster 0
Cluster 1
Cluster 2
Cluster 3
Cluster 4
Cluster 5
Cluster 6
Cluster 7
Cluster 8
Cluster 9

Results cont’d

- The chart representing all thirty clusters shows how many images of each sample contained animals.
- The purpose of this is to detect which of the models could return the most relevant results that show any trace of wildlife.
- For each of the models previously pictured, I used three of the “best” cluster samples that achieved my goal most thoroughly, wherein five sample images are pictured to be in each cluster. I then counted all of the images containing animals for each cluster sample and recorded them onto the chart (e.g., 3/5 sample images in Cluster 4 of VGG16 contained wildlife)

Analysis

- Even though the latter two models, ResNet50 and EfficientNetB0, handled clustering the most efficiently, they weren't as useful in achieving the goal as VGG16 was.
- Does this mean progress? VGG16 delivered more variety in the results, as expected, but still returned the most relevant results, as the animal sightings were the most consistent.
- Meanwhile EfficientNetB0 succeeded in categorizing the clusters, where each one had the most similarities in visual patterns. But there were very few animals to be spotted in the sample images of each cluster with this model.
- When it comes to unsupervised learning, variety may just be the best bet.

Brief Disclosure: Importance of CLEAN DATA!

- “Clean” data, while not always achievable, will deliver the best results — providing best accuracy and potential for insights.
- Some suggestions for improved performance:
  - Edge detection (to identify animals more efficiently)
  - Texture descriptors
- Perhaps using unsupervised learning could also improve the content of data sets themselves by filtering out the outliers, thus refining the quality of data collected.

Conclusion

- What I’ve learned:
  - Unsupervised clustering can help analyze population trends of wildlife by making data set filtering an easier process, allowing for less manual labor when recording statistics.
  - There is still much work to be done. Labeling is important when identifying specific species, and unsupervised learning can facilitate the classification process by recognizing similarities of images.
  - The three models I’ve used to demonstrate this performed adequately at recording the presence of wildlife in captured images. Depending on what type of results one deems most helpful for the situation, EfficientNetB0 could be the most useful compared to VGG16!
  - Seeing as the data set used was a challenging one with so many empty, blurry, and dark images containing wildlife, there is much hope for the accuracy of unsupervised clustering, as a good amount of the resulting clusters contained accurate images that were harder to interpret.

Future Directions?

- Using different types of clustering!
  - With more resources at my disposal, I would have loved to try experimenting with other types of clustering besides K-Means (e.g., DBSCAN, Gaussian Mixture Models).
- Working with more images
  - While working with the ENTIRE Snapshot Serengeti data set would have been overly-ambitious (again, ~7 million images), it would have provided me with better and more diverse data in the end.
- More direct animal identification
  - Even though the models can pick up similar patterns to group animals together, it would be useful to have a refined version of the project that returns more images of animals than the actual output had.