Using Deep Learning to Classify Animal Tracks

Project by Aryan Shah, Brinda Asuri, Haden Loveridge, Sam Chen, Sarah Dominguez, Kimble Horsack

Introduction:

Automated animal track identification is a critical tool for researchers and conservationists who rely on accurate species classification to monitor biodiversity and track animal movement in their natural habitats. Traditionally, experts have had to painstakingly analyze each footprint through visual inspection and comparison, a process that is both time-consuming and prone to human error. With the advent of deep learning, however, we now have the ability to train models on large datasets of animal track (footprints) images, enabling these systems to learn intricate patterns and subtle differences between species. This approach can significantly reduce the workload on researchers, allowing for faster, more cost-effective, and scalable wildlife monitoring programs.

Deep learning models, particularly convolutional neural networks and vision transformers, are uniquely suited for this task because of their ability to automatically extract and learn high-level features from raw images. These models can discern complex patterns in the texture, shape, and spatial arrangement of animal tracks, which are often too subtle for manual identification. By training on diverse datasets, the model becomes more robust to variations in image quality, lighting conditions, and environmental factors, ultimately delivering a more reliable classification system in real-world scenarios.

Moreover, integrating such technology into fieldwork not only expedites the identification process but also enhances the accuracy and repeatability of the results. Automated systems can operate continuously without fatigue, ensuring consistent monitoring even in remote or harsh environments. This consistency is crucial for long-term ecological studies where continuous data collection is necessary to track changes in wildlife populations over time.

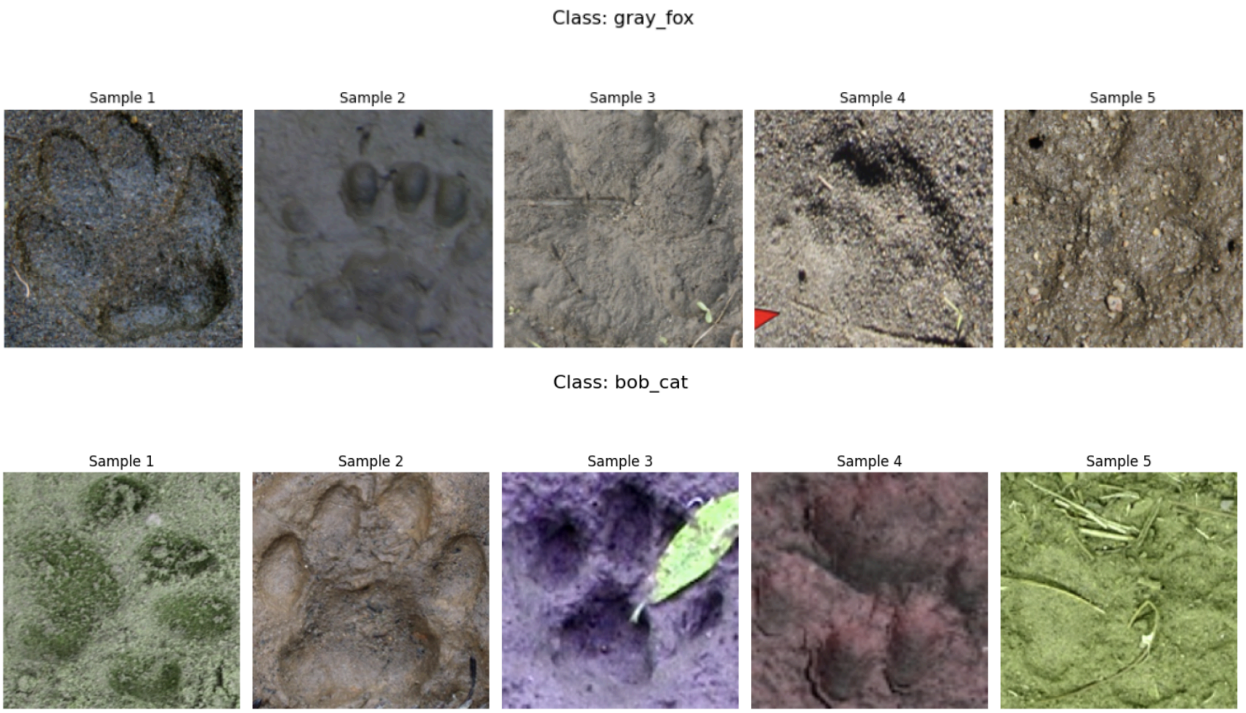
Data:

The dataset used in this project, the *Open Animal Tracks* dataset, was obtained through images manually collected by animal researchers in various field conditions. The images primarily represent mammals, along with several avian species, ensuring diverse coverage across different animal types. In total, the dataset contains 18 distinct species, specifically: Beaver, Black Bear, Bobcat, Coyote, Elephant, Goose, Gray Fox, Horse, Lion, Mink, Mouse, Mule Deer, Otter, Raccoon, Rat, Skunk, Turkey, and Western Gray Squirrel.

Overall, the dataset comprises 2,514 training images, 346 validation images, and 719 test images. Given the limited size of the dataset, we utilized pretrained models trained on large-scale image datasets to enhance the predictive capabilities of our final classification model.

To ensure robustness, footprints from each species were captured across multiple environmental conditions such as mud, dirt, gravel, snow, and sand. This variation helps the model accurately classify footprints based solely on their shape, independent of weather conditions or seasonal changes. Additionally, data augmentation techniques like image jitter were applied to randomize color conditions, and the orientation of the images was randomized to further enhance the model's generalization ability.

A notable challenge is the similarity between footprints of unrelated species; for instance, the gray fox and bobcat footprints appear remarkably alike despite belonging to entirely different suborders. Addressing such challenges was central to developing an accurate and reliable classification model.



Data processing:

We apply five primary transformations to our image data: resizing, randomizing the image orientation, applying random color jitter, converting the images to tensors, and finally normalizing the tensor values. First, all images are uniformly resized to 224 x 224 pixels, ensuring consistency in the input dimensions and reducing computational load during training. To help our model recognize tracks regardless of orientation, we apply random rotations

(between -30° to 30°) and random horizontal and vertical flips. These orientation-based augmentations significantly enhance the model's ability to generalize beyond the training dataset.

Additionally, we enrich the dataset by applying random color jitter, altering brightness, contrast, saturation, and hue to increase robustness against lighting and environmental variations. After these augmentations, the images are converted into tensors, a format suitable for deep learning frameworks. Lastly, we normalize these tensors using standard ImageNet RGB mean values [0.485, 0.456, 0.406] and standard deviations [0.229, 0.224, 0.225]. This normalization step ensures numerical stability and accelerates training convergence, ultimately improving the accuracy and reliability of our track-classification model.

ResNet 18:

For our initial model training, we decided to use an out-of-the-box deep learning model instead of building one from scratch. After evaluating several candidate models, ResNet-18 stood out due to its balance between complexity, accuracy, and computational efficiency. ResNet-18 contains about 12 million trainable parameters and is structured into multiple convolutional blocks paired with skip connections (also known as residual connections). These skip connections allow the network to efficiently handle issues like vanishing gradients, making it easier to train deeper networks that can effectively capture intricate visual patterns.

A major advantage of leveraging a pre-trained ResNet-18 is its inherent sensitivity to fundamental visual elements such as edges, textures, and basic geometric patterns that are features consistently observed across natural images. Since it has been pre-trained on a large-scale dataset (like ImageNet, with millions of images), the model already possesses generalized feature-extraction capabilities. Thus, fine-tuning this model for our specific task of animal footprint classification significantly reduces the training time and data required.

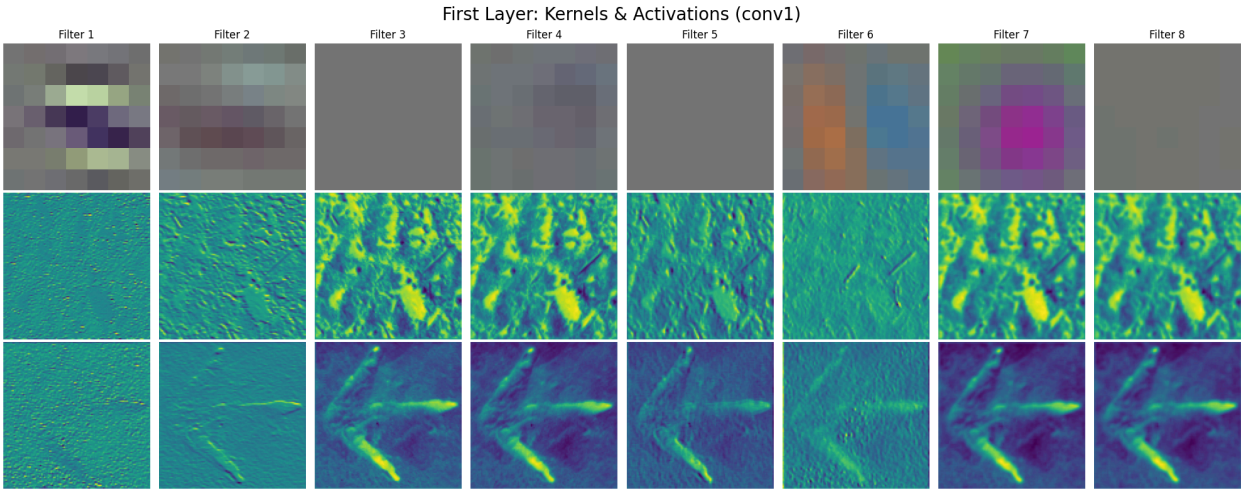
Since our goal is to classify images into 18 distinct animal categories, we specifically modified the final fully connected layer of the original ResNet-18 model, which was initially designed to distinguish between 1,000 classes. By replacing this layer, we tailored the model’s outputs directly to our set of animals so it can better align with our unique classification needsThis customization retains the powerful feature extraction capabilities of ResNet-18, allowing it to efficiently adapt to our unique classification task and improving overall predictive accuracy.

Visualizing the filters:

To interpret how our convolutional neural network (ResNet-18) distinguishes between different animal tracks, we analyzed the model’s internal components—specifically the kernels (filters) and their resulting activation maps. Kernels are small feature detectors that slide across the image and learn to highlight specific patterns like edges, textures, or complex shapes. When

applied to an image, a kernel produces an activation map that shows where in the image the kernel "responded" most strongly.

Below are the original input images used in this analysis: one of a lion footprint and one of a turkey footprint. These two species represent very different foot structures, which the model must learn to distinguish.

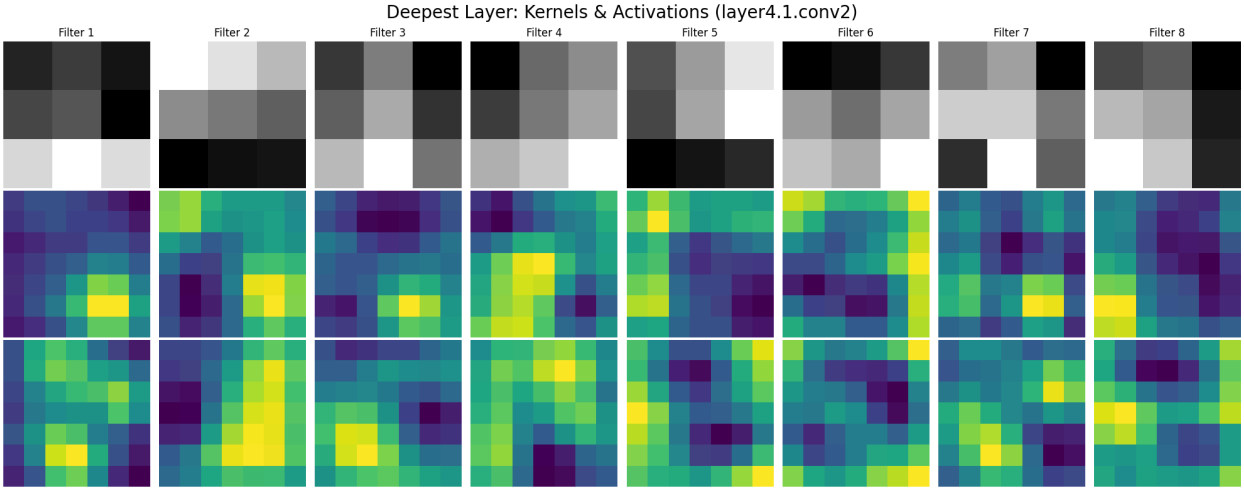


In this first figure, we display the first 8 kernels from the model's first convolutional layer (conv1) and their corresponding activations for the lion and turkey images.

- Top row: Learned RGB filters, each capturing simple features such as horizontal or vertical edges, textures, or color contrasts.
- Middle row: Activation maps for the lion image.

- Bottom row: Activation maps for the turkey image.

At this early stage, we can still visually recognize the footprint structure in the activations. Many filters highlight general features like ridges or outlines of the toes. Interestingly, the activations for both animals look quite similar—this is expected since early layers are meant to detect shared low-level features, not yet animal-specific traits.



The next figure visualizes the first 8 kernels from the deepest convolutional layer (layer4.1.conv2), which sits right before the classification output.

- Top row: These filters are no longer human-interpretable shapes but encode highly abstract patterns.
- Middle row: Activation maps for the lion image.
- Bottom row: Activation maps for the turkey image.

Here, we can no longer see clear outlines of the tracks—activations are abstract and sparse. However, this is exactly what we expect from a well-trained deep network: the model has transformed the input into high-level, animal-specific representations. The differences between the two species are now clearly reflected in the activation maps—certain filters fire only for the lion, while others respond strongly to the turkey.

This layered visualization reveals how the model transforms raw image data into meaningful features:

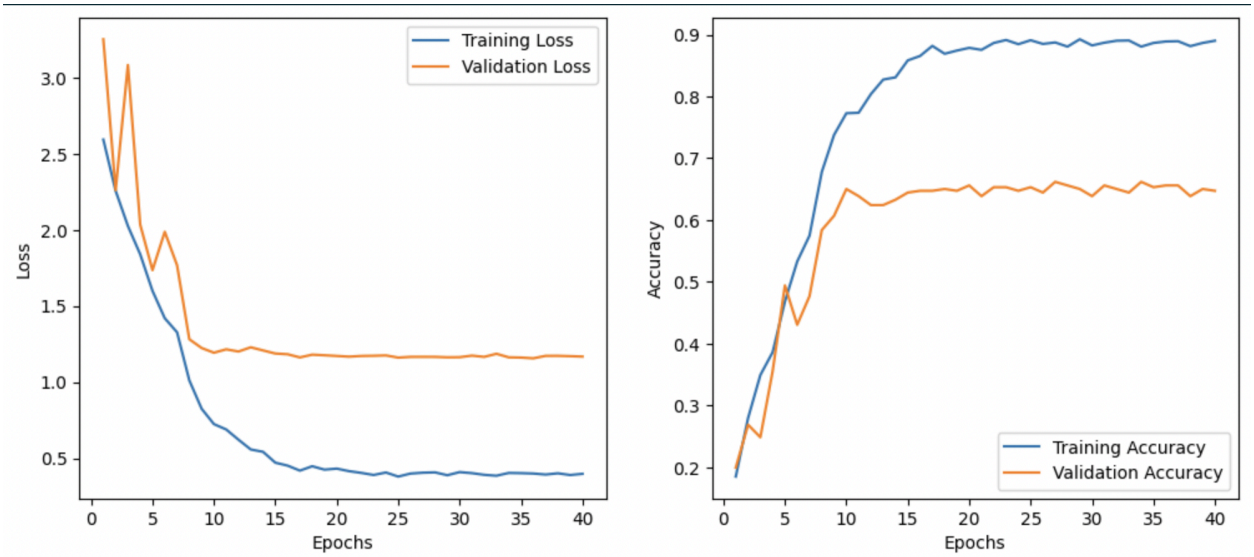
- Early layers learn generic visual cues that are similar across species.
- Deeper layers capture complex, animal-specific traits, enabling accurate classification.

- The progression from recognizable edges to abstract patterns mirrors how the model builds its understanding from the ground up.

By visualizing both the learned kernels and their activation maps, we gain insight into what the model has truly learned—and confidence that it is distinguishing footprints for the right reasons.

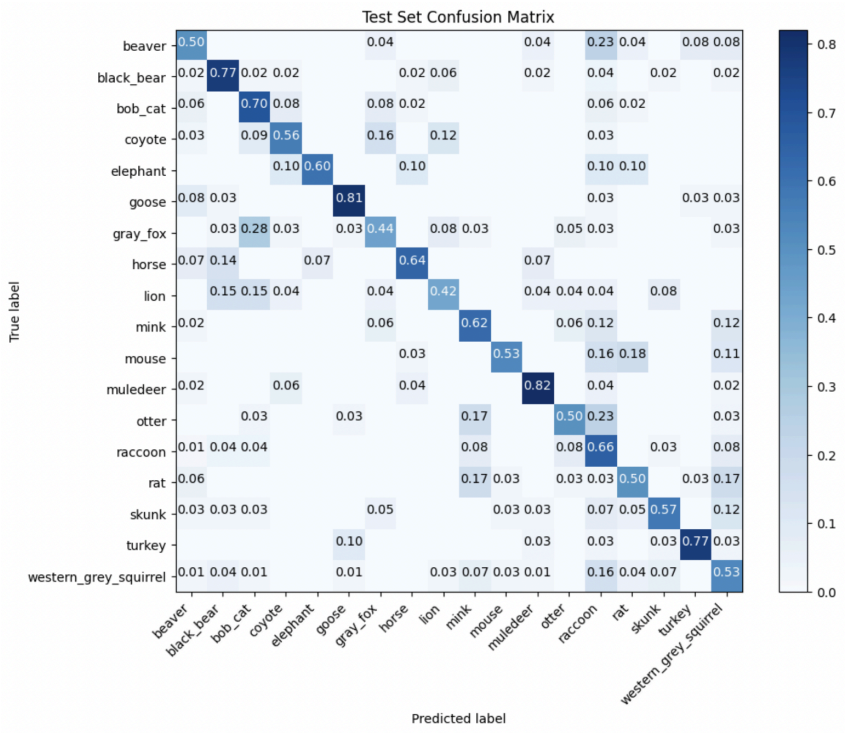
Results:

After training our ResNet-18 model over 40 epochs, we observed that the model's classification accuracy reached a plateau, suggesting limited improvements beyond a certain point. This could indicate either the limitations of our relatively small dataset or that the model had already captured most of the meaningful features distinguishing these animal footprints.



The three animal classes with the highest classification accuracy were Mule Deer (0.87 accuracy), Goose (0.83 accuracy), and Turkey (0.78 accuracy). It's understandable that these species were classified more accurately, as their footprints have distinct characteristics. For example, goose and turkey footprints are avian, featuring very different shapes and arrangements compared to most mammalian tracks, making them easier for the model to distinguish.

Interestingly, elephants, known for having quite distinct and large footprints, achieved a surprisingly modest accuracy of 0.60. Despite the uniqueness of an elephant footprint's size and roundness, it seems the model occasionally confused them with footprints from other mammals, potentially carnivorous species, which can also exhibit rounded silhouettes depending on environmental factors such as mud or snow conditions.



Additionally, the confusion matrix revealed the highest confusion occurred between Gray Fox and Bobcat, with an accuracy of only 0.28 for correctly classifying these two species. This confusion aligns closely with real-world observations, as these footprints are extremely similar despite belonging to entirely different suborders. Such results highlight the nuanced detail required to distinguish between similar mammalian tracks, suggesting the model, much like human experts, needs to carefully identify subtle features.

Future explorations could include investigating whether the model encounters similar challenges when differentiating between other species pairs with visually similar tracks, including avian species, or implementing additional data augmentation techniques and increasing dataset size to boost accuracy further.

Visual Transformer:

Leveraging transfer learning with a Vision Transformer (ViT) offers a powerful approach for classifying animal tracks, and our project demonstrates an effective end-to-end deep learning pipeline using this method. Our process begins with data preparation, where we apply a series of image transformations to standardize inputs for the ViT model. Images are resized to 224×224 pixels, converted to PyTorch tensors, and normalized using ImageNet's standard mean and

standard deviation values. This preprocessing aligns our data with the distribution the pretrained model expects, thereby enhancing performance during fine-tuning. We then organize our data into training, validation, and test sets using a custom dataset class, AnimalTracksDataset, and load them efficiently with DataLoaders. With a batch size of 32 and appropriate shuffling for training data, the DataLoaders ensure smooth and effective feeding of data into the model during training.

Our model is built using the Vision Transformer architecture, specifically the 'vit_base_patch16_224' model provided by the timm library. This model divides each image into fixed-size patches, processes these patches as a sequence, and applies self-attention mechanisms to capture long-range dependencies. The pretrained weights, originally learned on the large-scale ImageNet dataset, enable the model to extract rich and robust features, such as edges and textures. By replacing the final classification head with a new fully connected layer tailored to the number of classes in our dataset, we adapt the pretrained model to our specific task of classifying animal tracks. The model is then moved to the appropriate device, using a GPU if available to accelerate computations.

Training is driven by a well-structured pipeline that leverages a standard set of deep learning components. We use CrossEntropyLoss, which is ideal for multi-class classification tasks, as it quantifies the discrepancy between the predicted probabilities and the true labels. The Adam optimizer, set with a learning rate of 1e-4, is employed for its adaptive learning rate properties that facilitate faster convergence, especially during fine-tuning. Additionally, a StepLR scheduler is used to reduce the learning rate by a factor of 0.1 every 5 epochs, allowing the model to make finer adjustments as training progresses. The training process itself is organized into epochs that alternate between training and validation phases. During training, the model processes each batch of images, computes predictions, calculates loss, and updates its weights using backpropagation. Metrics such as loss and accuracy are continuously monitored, and if a new best validation accuracy is reached, the model's state is saved.

After completing the training, we visualize the loss and accuracy curves over the epochs using matplotlib, which helps us diagnose potential issues such as overfitting or underfitting. For evaluation, the model is tested on a separate test set where its predictions are compared against the ground truth labels to calculate overall accuracy. Furthermore, we generate a confusion matrix using sklearn's confusion_matrix function to gain detailed insights into how well the model performs on each class, highlighting areas where certain classes may be confused with others.

The outcomes of our approach are promising; although the model achieves a moderate overall accuracy, the results underscore the efficiency of transfer learning with a Vision Transformer. The pretrained model's robust feature extraction capabilities are effectively fine-tuned to the specific nuances of animal tracks, allowing the network to learn relevant patterns quickly even

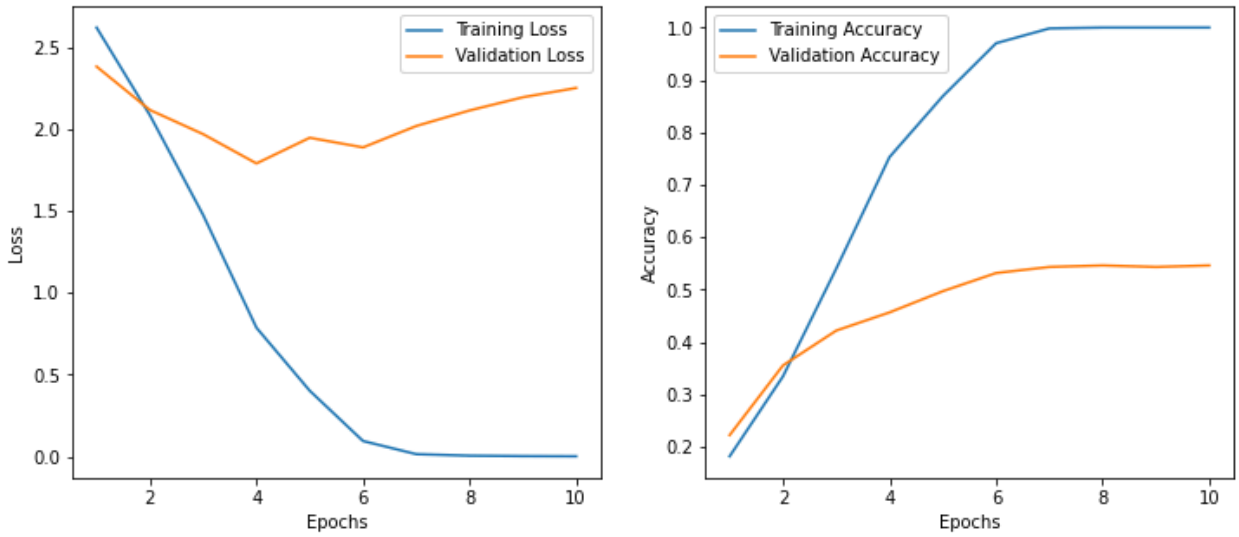
with a relatively small dataset and a limited number of training epochs. However, there remains room for improvement, as factors like the short training duration, potential hyperparameter adjustments, and the incorporation of more aggressive data augmentation techniques could further enhance performance.

In summary, our project illustrates how leveraging a pretrained Vision Transformer can streamline the process of adapting a state-of-the-art deep learning model to a domain-specific task such as animal track classification. The combination of rigorous data preprocessing, thoughtful model customization, and a carefully tuned training process demonstrates the practical benefits and challenges of transfer learning. This approach not only reduces training time and computational resource requirements but also provides a solid foundation for further refinements that could lead to even better performance in future work.

Since our results from this model were not ideal, we tried to enhance our model. We incorporated a progressive freezing strategy to further refine the training process. Our model, initially, had all layers of the model frozen to prevent any weight updates, and only the classification head and the last two transformer blocks were unfrozen. This approach allows the model to start by learning high-level features relevant to our specific task, while preserving the robust, pretrained representations in the lower layers. After a predefined number of epochs, specifically after epoch 5, an additional transformer block is unfrozen, and the optimizer is reinitialized so that it now updates the newly trainable parameters. This gradual unfreezing is intended to stabilize the early training phase by relying on the reliable features of the pretrained model, and then slowly allowing deeper layers to adjust to the new data. Although the model using progressive unfreezing achieves an accuracy in a similar range, this technique opens up avenues for more refined adaptation and potentially better performance over extended training durations.

Overall, both approaches illustrate the practical benefits of transfer learning with Vision Transformers. While the standard fine-tuning approach leverages the robust pretrained features directly by replacing the classification head, the progressive unfreezing strategy adds a layer of control over the training process, preserving foundational features in the early stages and gradually adapting deeper layers. This balance between preserving learned representations and allowing new adaptations is key to optimizing performance, especially when dealing with complex tasks like animal track classification. The insights gained from training curves, accuracy metrics, and confusion matrix analyses guide further refinements, such as extended training, enhanced data augmentation, and hyperparameter tuning, ultimately paving the way for even better future performance.

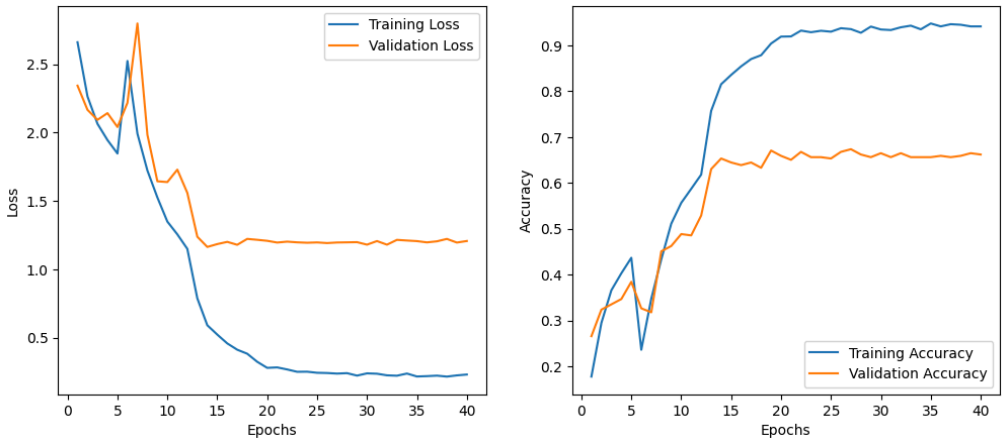
Results:



It’s obvious from the results that the first transformer model overfits heavily to the training data. This is shown by the training and validation curves diverging from one another. In a deep learning context, addressing overfitting is essential for building a robust model. Additionally for the first model, our accuracy resulted in 60% after epoch 5, where the loss function stopped improving. We tried to improve this using freezing techniques, but that ended in a lower accuracy level of 58%. Techniques such as dropout and weight decay could help mitigate overfitting. However, we think that the size of our dataset does not support these techniques as a real solution.

Experimental Methods:

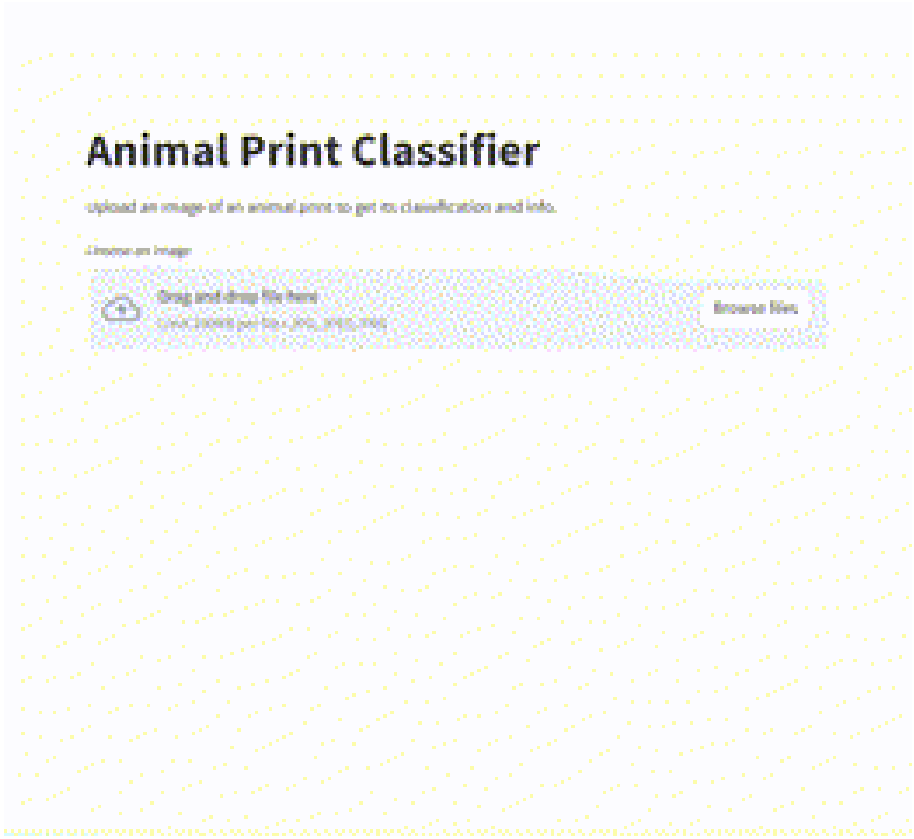
Our team also experimented with progressive freezing with CNN. In the ResNet model, we initially froze portions of the network and then unfroze and trained the entire network every five epochs. The goal was to determine if training the full convolutional and max-pooling layers would enhance the model's ability to learn patterns in our dataset. However, the results were mediocre, with performance remaining around 65%, similar to the baseline ResNet model.



Interactive Components:

As part of practically applying our work, we also developed an interactive application that allows users to simply upload a photo of an animal print. Utilizing Streamlit, a user-friendly platform for building data-driven web apps, our solution allows anyone, from seasoned researchers to casual wildlife enthusiasts, to effortlessly upload a photo to the website of an animal footprint. Then, upon receiving the uploaded image, our integrated pre-trained ResNet-18 model immediately analyzes it and accurately predicts the animal species. Furthermore, the application enriches user experience by providing insightful information, including typical habitats, geographic locations, and engaging fun facts about the identified species.

This approach has practical applications in the field, making it an invaluable tool for researchers, conservationists, and wildlife enthusiasts alike. With the ability to capture animal prints on the go, users can instantly learn more about the species they encounter. The app not only aids in accurate and efficient species identification but also engages users by sharing interesting, educational insights about each animal, thereby supporting both scientific research and broader public awareness.



GIF of User Interface pictured above.

Future Scope:

While the models were able to show decent results with the data provided, there remains significant potential for further improvement by expanding and diversifying our dataset. Increasing the number of images, particularly from a broader range of species, would likely enhance the overall accuracy and robustness of our models. A more varied dataset would expose the model to a wider spectrum of features and patterns, allowing it to better distinguish between subtle differences among species.

Another important consideration is the variety of image perspectives. Most of the images used during modeling were top-down shots, which are advantageous for capturing clear contours and shapes critical for identifying the overall structure of tracks. However, these images inherently lack depth information and may not fully represent the three-dimensional aspects of animal footprints. In future iterations, it would be beneficial to include images captured from various angles, such as oblique or side views, as these could offer richer detail about depth, texture, and environmental context. Incorporating diverse perspectives could provide complementary information, helping the model better differentiate between similar species and enhancing its generalization capabilities across different conditions and environments.

Furthermore, incorporating diverse imaging conditions, such as different lighting, weather conditions, and seasonal variations could further enhance model performance. This would allow the model to learn more robust features that are invariant to external factors, thereby improving its accuracy in real-world applications.

Lastly, one significant area of improvement would be transitioning our current web-based interface into a dedicated mobile application. A mobile app would significantly enhance accessibility by allowing users to directly capture and upload images from their smartphones. This convenience is particularly beneficial for researchers and students conducting fieldwork, enabling them to instantly identify tracks on-site without the need for separate camera devices or transferring images between platforms. Such a mobile solution would promote more widespread and practical usage, facilitating faster data collection and greater overall user engagement.