Dataset Impact on Image Classification Models

Artificial Intelligence Progress Report

Narindra Balkissoon – CSC 547 – UT2

Quinn Sturm – CSC 547 – UT2

Chang Wang – CSC 547 – UT2

Shuyun Shen – CSC 447 – UT1

Spring 2025
Dr. Lina Chato

Abstract

In this project, we aim to repurpose the top-performing model architecture from last semester's computer vision project which used deep learning to classify , specifically those based on VGG16 and ResNet50. We will retrain these models on a new dataset focused on wild big cat species. Our primary objective is to evaluate the generalization capabilities of these models when applied to a more complex, real-world dataset. Additionally, we will identify areas for improvement through hyperparameter tuning and explore alternative pretrained models to optimize performance further. Through systematic experimentation and comparison with previous results, we aim to validate the adaptability and robustness of our earlier models across different image classification tasks.

Introduction and Literature Review

Convolutional Neural Networks (CNNs) have shown exceptional performance on structured and curated datasets. However, classifying images in real-world scenarios presents additional challenges, including background clutter, variations in lighting, occlusions, and significant differences within the same class. Building on the Dog Classification project from last semester, we now aim to test the adaptability of our previous architectures by applying them to a new domain: classifying wild big cat species using the "10 Big Cats of the Wild" dataset.

This dataset is notably more complex than typical academic datasets, featuring images captured in natural environments with uncontrolled backgrounds, varying lighting conditions, and different animal poses. This complexity increases the diversity within classes and the similarity between different classes, making the classification task significantly more challenging. Our project addresses two key questions: Can the models developed previously generalize to this new and difficult context? What adjustments or enhancements might be necessary to optimize their performance?

Our project focuses on assessing how effective it is to transfer existing convolutional neural network (CNN) architectures, specifically ResNet50 and VGG16, from dog image classification to the classification of big cats. Each selected paper offers valuable insights related to our approach, including architectural considerations, transfer learning strategies, and practical training methods.

Nguyen et al. (2017) developed a CNN-based automated wildlife recognition system using camera trap images, achieving impressive classification accuracy, 96.6% for detecting animal presence and 90.4% for identifying specific species. Their research demonstrates that CNNs can effectively generalize for wildlife identification tasks when trained on diverse animal datasets. In a similar vein, our project uses CNNs for animal species classification, but we specifically investigate whether models originally trained on dogs can also generalize to big cats. Unlike Nguyen et al.'s study, our focus is on transfer learning between closely related animal categories rather than building a CNN system from scratch.

Simonyan and Zisserman (2015) introduced VGG architectures characterized by deep stacking of small convolutional filters (3x3). Their work demonstrated that network depth significantly improves the accuracy of CNN models on large-scale image recognition tasks. This directly aligns with our use of VGG16, offering foundational knowledge about why deeper CNN architectures perform better. Our project specifically leverages these insights to test VGG16's effectiveness when adapted to a moderately sized, closely-related dataset, focusing on how well these depth-related advantages persist in a transfer learning context between dog and big cat images.

He et al. (2015) presented ResNet architectures that solved the degradation problem of deeper networks through residual learning. Their research demonstrated significant performance improvement on ImageNet by training networks as deep as 152 layers efficiently. Our project employs ResNet50, directly benefiting from their findings about residual learning advantages. While He et al. focused primarily on depth and optimization in general contexts, we specifically tested ResNet's transferability, examining whether residual learning offers clear benefits when shifting from dog classification tasks to big cat classification tasks, something their original paper did not explicitly investigate.

Kornblith et al. (2019) conducted a systematic study on transfer learning across various ImageNet architectures. Their findings demonstrated a strong correlation between improved performance on ImageNet and enhanced transfer accuracy. They highlighted the critical importance of hyperparameter tuning and regularization for successful transfer learning. This aligns closely with our project's methodological approach, particularly our focus on adjusting hyperparameters such as learning rate, batch size, and dropout. We apply their insights by experimenting with hyperparameter tuning to optimize performance when transferring knowledge from dog classification to big cat image classification. Unlike their broader analysis across multiple datasets, we concentrate specifically on closely related animal categories to extract more precise practical insights.

Sabottke and Spieler (2020) investigated how input image resolution affects the performance of convolutional neural networks (CNNs) in radiographic image classification tasks. Their research highlights important trade-offs between computational efficiency and accuracy. Although their focus on medical imaging differs significantly from our project, their findings on the impact of image preprocessing decisions are relevant, especially when considering our choices regarding image resolution during preprocessing. In our project, which involves a moderately sized dataset of big cats, we also face a trade-off between computational efficiency and accuracy. However, unlike Sabottke and Spieler, we do not systematically explore the entire range of resolution-performance options. Instead, we select a resolution based on their insights to optimize our model's generalization performance.

In summary, these five papers collectively offer strong theoretical and methodological frameworks for our project. We specifically apply their findings regarding model architecture,

hyperparameter tuning, and practical training considerations. However, our study uniquely builds on these foundational insights by explicitly testing the transferability of models between two closely related yet distinct animal datasets: dogs and big cats. This focus provides clearer practical implications regarding architectural adaptability, transferability, and optimization strategies.

Data and Data Preparation

The dataset used in this project, titled "10 Big Cats of the Wild," was obtained from Kaggle. It includes a total of 2,439 images spread across 10 categories: African Leopard, Caracal, Cheetah, Clouded Leopard, Jaguar, Lions, Ocelot, Puma, Snow Leopard, and Tiger. These images showcase the animals in various real-world conditions, featuring different habitats, lighting situations, and poses, which provide a realistic basis for model evaluation.

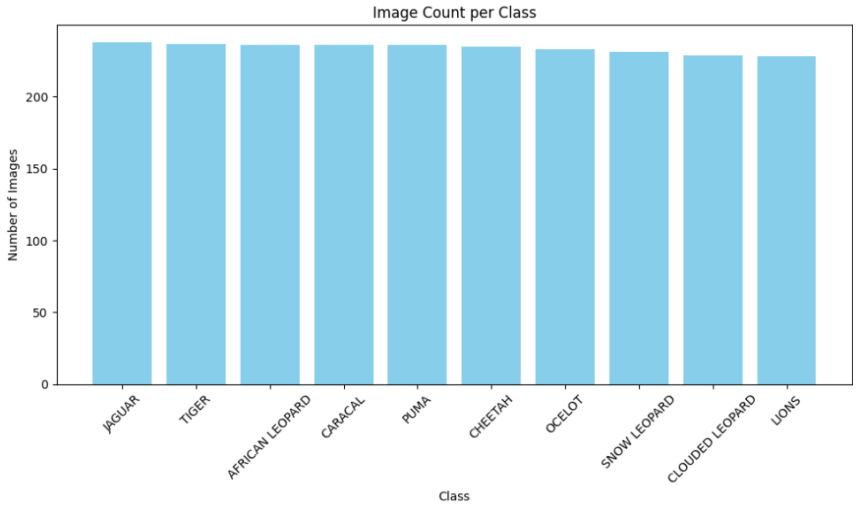

Figure 1 represents the distribution of each class in the data set

	Class	Image Count	Percentage
7	JAGUAR	238	10.175289
4	TIGER	237	10.132535
9	AFRICAN LEOPARD	236	10.089782
3	CARACAL	236	10.089782
6	PUMA	236	10.089782
0	CHEETAH	235	10.047029
8	OCELOT	233	9.961522
2	SNOW LEOPARD	231	9.876015
5	CLOUDED LEOPARD	229	9.790509
1	LIONS	228	9.747755

.

Figure 2 represents the details of each class in the dataset

To prepare the data for training, all images were resized to 224×224 pixels to meet the input size requirements of the pretrained models. Various data augmentation techniques were employed to enhance the variability and robustness of the training set. These techniques included random rotations of up to ±20 degrees, horizontal flipping, adjustments to brightness and contrast, and zoom transformations. Additionally, pixel values were normalized to the [0, 1] range to facilitate faster convergence during training. The dataset was divided into three parts: 60% for training, 30% for validation, and 10% for testing. The data set we're using is far more diverse and balanced than the Stanford dog breed dataset we used. The images have also been cropped for us. However, the data set only contained 10 classes which may have contributed to the overall accuracy. We also tried another data set "big-cats-images-dataset" which also contained 9 classes, but there weren't enough images for each class.

Methodology

For this project, we will utilize transfer learning by starting with the pretrained VGG16 and ResNet50 models that have been trained on ImageNet. We will remove their original fully connected layers and replace them with a new classification head. The ResNet50 head will include a Global Average Pooling layer, one or more dense layers with ReLU activations, a dropout layer with a dropout rate of 0.5 for regularization, and a final softmax output layer designed for 10-class classification. This can be seen in figure x below. We changed the output layer from 120 to 10, to match the classes in the data set, this is not present in figure x yet, as we plan to redraw the architecture. Currently the last 30 layers in addition to the custom layers of the ResNet50 model are trainable.

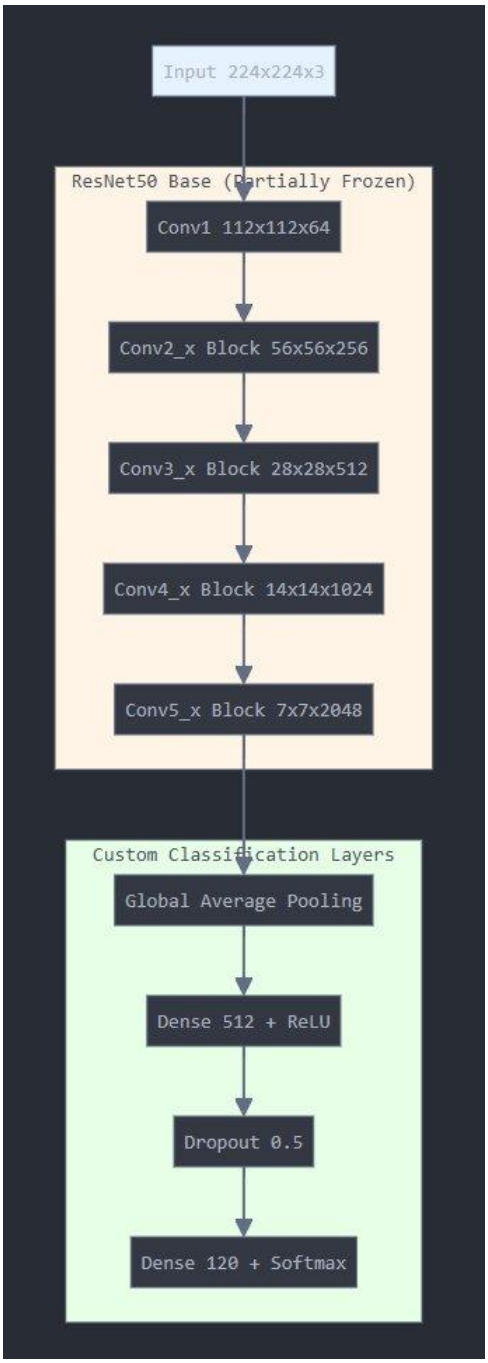

Figure 3 represents the model design for the ResNet50 used in the previous project

Training will be conducted in two phases. In the first phase, known as feature extraction, all convolutional layers of the pretrained models will be frozen, allowing only the new classification head to be trained. This process enables the model to quickly adapt high-level learned features to the new task without interfering with the generalized lower-level features. In the second phase, which is called fine-tuning, we will selectively unfreeze the last few convolutional blocks and retrain the model using a much lower learning rate. This approach fine-tunes the feature

representations to better match the characteristics of the wild cat images while preserving the general knowledge obtained from ImageNet.

Hyperparameters will be set based on prior experience. The learning rate will begin at 0.001 during the feature extraction phase and will be reduced to 1e-5 during fine-tuning. Batch sizes will range from 32 to 64, depending on the available GPU memory. We will start with the Adam optimizer, but we will also evaluate the performance of SGD with momentum and RMSprop to assess optimizer sensitivity. Early stopping will be implemented to avoid overfitting as well as to save resources, monitoring validation loss and halting training if no improvement is observed over a predetermined number of epochs. Additionally, a learning rate scheduler will be employed to automatically reduce the learning rate when the validation performance plateaus.

To enhance generalization and prevent overfitting, we will implement regularization techniques such as dropout, data augmentation, early stopping, and, optionally, L2 weight decay. Data augmentation is particularly crucial due to the relatively small size of the dataset and the variability found in real-world images.

We also aim to perform the same experiments using our best VGG16 model from last semester. Which can be seen in figure x below. We added a customer layer which contains a flatten, dropout, dense, batch normalization, dropout, dense, batch normalization and finally a soft max dense layer for an output probability of 10 classes.

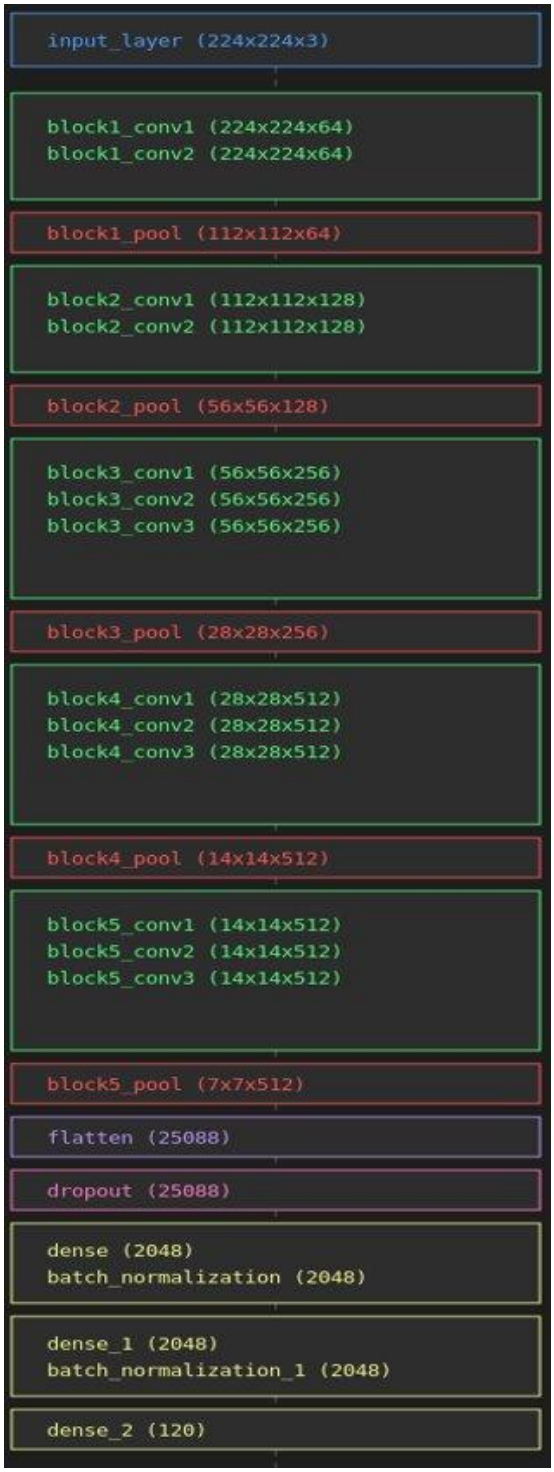

Figure 4 represents the model design of the VGG16 base model used in the previous project.

Experiments

To enhance generalization and prevent overfitting, we will implement regularization techniques such as dropout, data augmentation, early stopping, and, optionally, L2 weight decay. Data augmentation is particularly crucial due to the relatively small size of the dataset and the variability found in real-world images.

Further experiments will involve testing different train-validation-test split ratios, such as 80/10/10, to assess if the model benefits from a larger training set. We will also compare the impact of different optimizers on model convergence and final performance. Additionally, we will conduct experiments to evaluate how different levels and types of data augmentation affect model generalization on unseen test data.

Table 1 represents a list of experiments we have tried so far

Base Model	**Preprocessing Method**	**Model Alteration**	**Batch Size**	**Data Split**	**Optimizer**	**Epochs**
VGG16	OpenCV	None	64	80/20	Adam	42 with early stopping
VGG16	Data Augmentator	None	64	60/30/10	Adam, 0.0001	16 with early stopping
ResNet50	Data Augmentator	None	64	60/30/10	Adam, 0.001	39 with early stopping
ResNet50	Data Augmentation (same as VGG16)	None	64	60/30/10	SGD, 0.001	50, with early stopping
VGG16	Data Augmentation (same as VGG16)	All layers frozen, except custom layers	32	60/30/10	Adam, 0.0001	15 with early stopping

Expectations and Results

We propose that models utilizing transfer learning will significantly outperform those trained from scratch, particularly due to the relatively small size and complexity of the dataset. Among the baseline models, ResNet50 is anticipated to generalize better than VGG16 because of its advanced architecture and skip connections, which help reduce overfitting.

Proper hyperparameter tuning and suitable data augmentation are expected to be crucial in improving validation accuracy and reducing the gap between training and testing performance. Through fine-tuning, we anticipate achieving higher precision, recall, and F1 scores, particularly for challenging classes such as the jaguar and leopard, which are visually similar.

The results will be presented as training and validation accuracy, loss curves, confusion matrices to visualize misclassifications, along with per-class precision, recall, F1-scores, and comparative performance tables summarizing findings across different models and settings.

Table 2 represents the list of experiments and results so far

Model	**Preprocessing Method**	**Model Alteration**	**Batch Size**	**Data Split**	**Optimizer**	**Epochs**	**Test Accuracy(%)**
VGG16	OpenCV	None	64	80/20	Adam	42 with early stopping	90.40%
VGG16	Data Augmentation	None	64	60/30/10	Adam, 0.0001	16 with early stopping	92.24
ResNet50	Data Augmentation	None	64	60/30/10	Adam, 0.001	39 with early stopping	90.2
ResNet50	Data Augmentation (same as VGG16)	None	64	60/30/10	SGD, 0.001	50, with early stopping	93.88
VGG16	Data Augmentation	None	32	60/30/10	Adam, 0.0001	15 with early stopping	93.06

Experiment 1: VGG16 using OpenCV

For experiment 1 we used OpenCV to augment the images. We resized the images and performed color augmentation. We used a 80/20 split for the data having a training set and a test set only. Early stopping was implemented with a patience of 10, we restored the best weights and use reduce of learning rate of plateau with a factor of 0.1 and a patience of 10. The experiment ran for 42 epochs and restored the weights from the 32nd epoch. We achieved a loss of 0.32 and an accuracy of 90.4%.

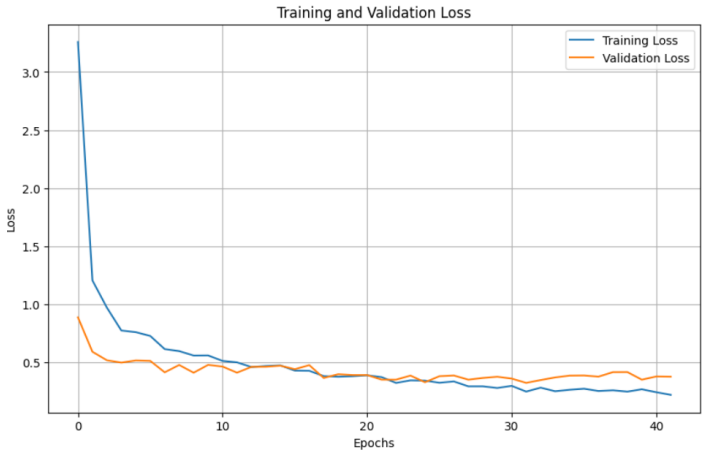

Figure 5 represents the training and validation loss curve for experiment 1

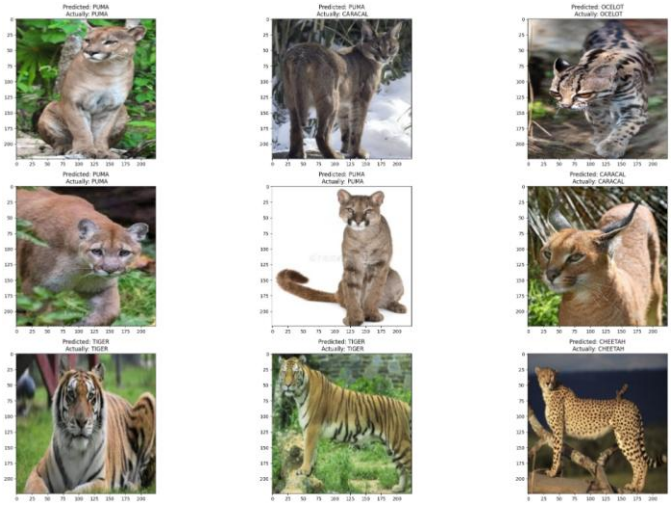

Figure 6 represents a sample of the predicted classes from the test set

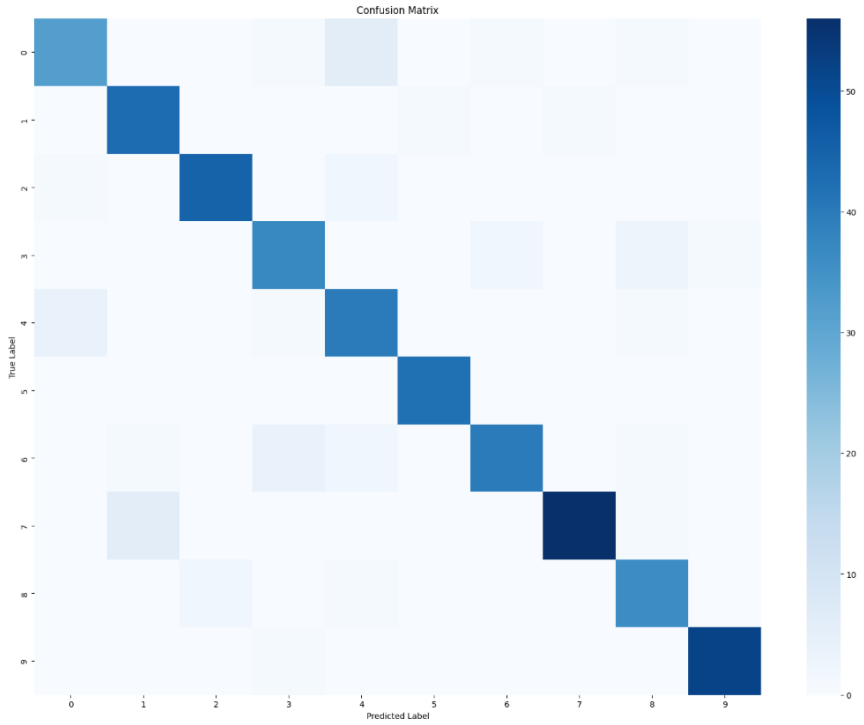

Figure 7 represents the confusion matrix for experiment 1

Experiment 2:

For experiment 2 we split the data into a 60/30/10 ratio for the training/validation/test sets. Then we used the image data generator from keras to augment the data as stated in the data section. In this experiment all the layers were frozen except the custom layers. Adam optimizer was used as well as a learning rate of 0.0001. Early stopping, reduce learning rate on plateau was used, same as in experiment 1. The training stopped at the 16th epoch at a validation loss of 0.3287. The accuracy of the test set was 92.24%.

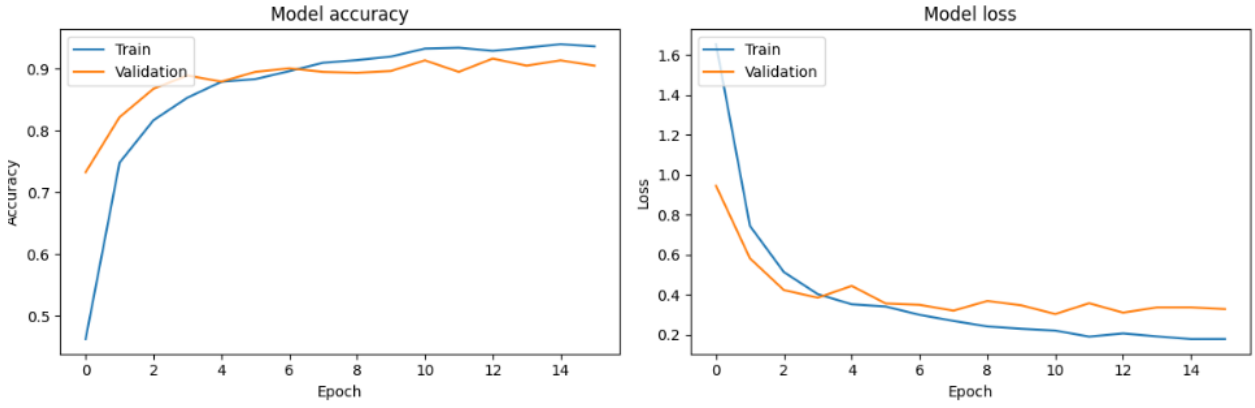

Figure 8 represents the accuracy and loss curves for experiment 2

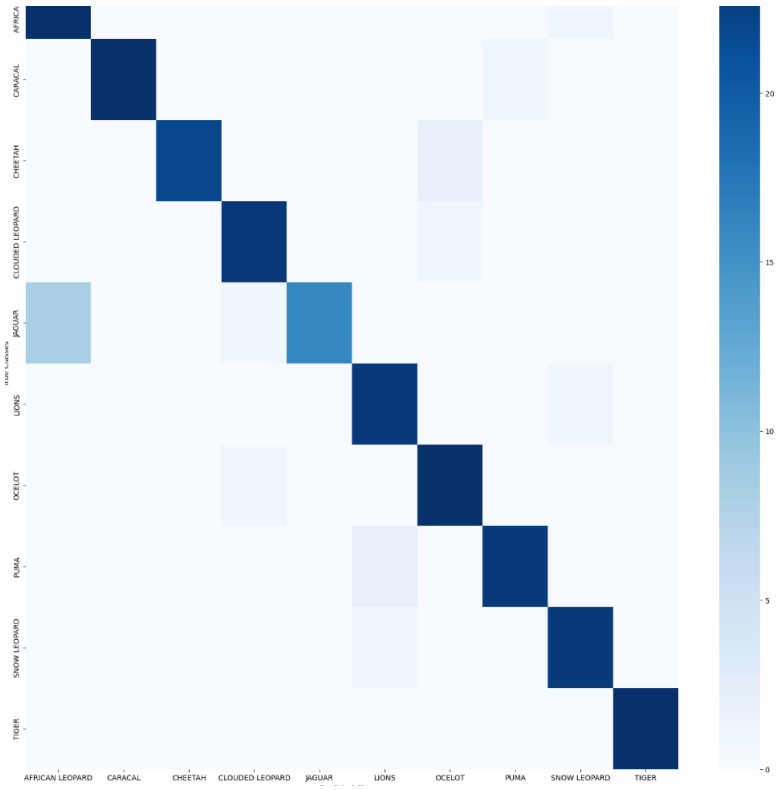

Figure 9 represents the confusion matrix for experiment 2

Figure 10 represents a sample of the predictions from experiment 2

Experiment 3:

In experiment 3 we used ResNet50, a learning rate of 0.001 was used with Adam as the optimizer. The same early stopping was used as well as reduce learning on plateau as the

previous experiment. All layers except the last 30 in the ResNet50 base model are frozen. The training stopped at the 39th epoch at a validation loss of 0.3348. The accuracy of the test set was 90.2%.

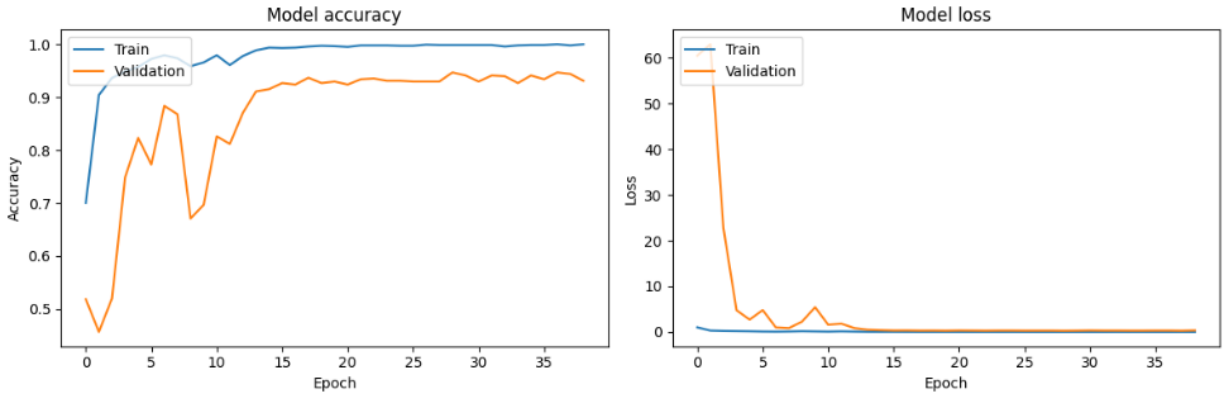

Figure 11 represents the accuracy and loss curve for experiment 3

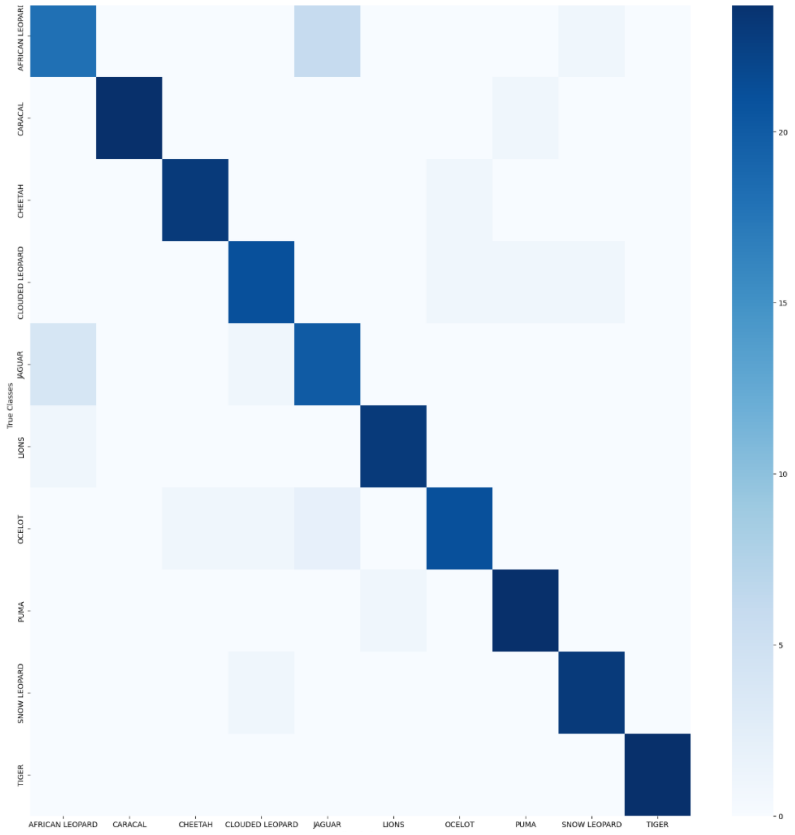

Figure 12 represents the confusion matrix for experiment 3

	precision	recall	f1-score	support
AFRICAN LEOPARD	0.78	0.72	0.75	25
CARACAL	1.00	0.96	0.98	25
CHEETAH	0.96	0.96	0.96	24
CLOUDED LEOPARD	0.88	0.88	0.88	24
JAGUAR	0.71	0.80	0.75	25
LIONS	0.96	0.96	0.96	24
OCELOT	0.91	0.84	0.87	25
PUMA	0.92	0.96	0.94	25
SNOW LEOPARD	0.92	0.96	0.94	24
TIGER	1.00	1.00	1.00	24
accuracy			0.90	245
macro avg	0.90	0.90	0.90	245
weighted avg	0.90	0.90	0.90	245

Figure 13 represents the classification report for experiment 3.

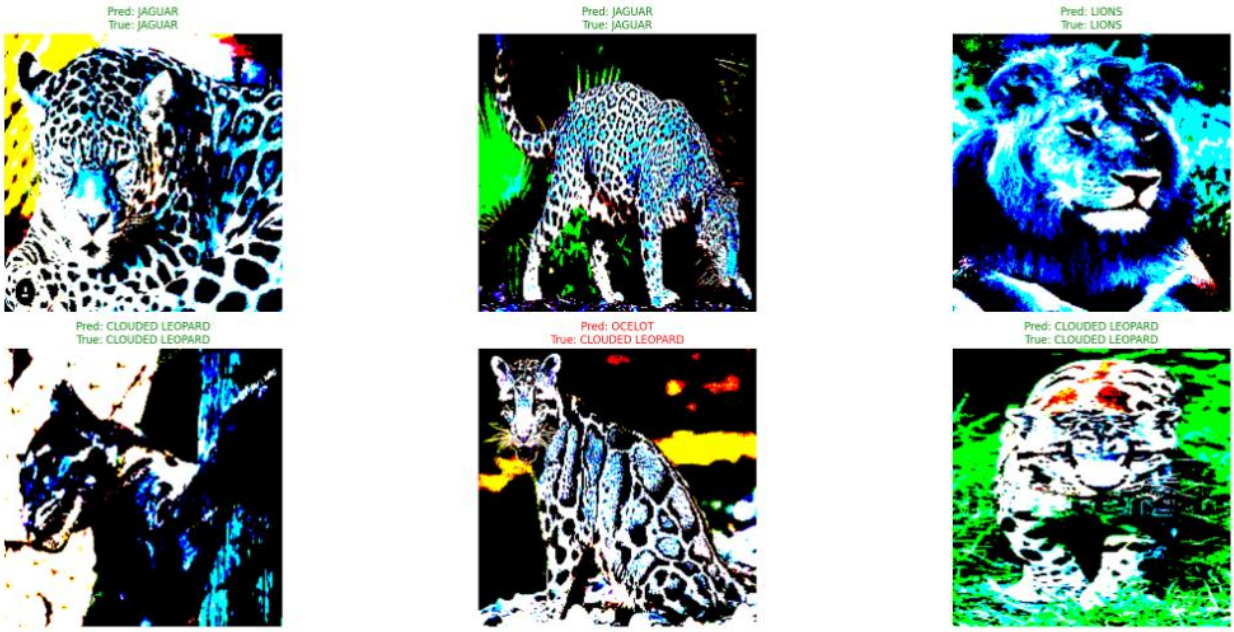

Figure 14 represents a sample of the predictions from experiment 3

Experiment 4:

Experiment 4 is identical to experiment 3, except instead of Adam, Stochastic Gradient Descent is used with a learning rate of 0.001. This experiment got a 93.88% accuracy on the test set.

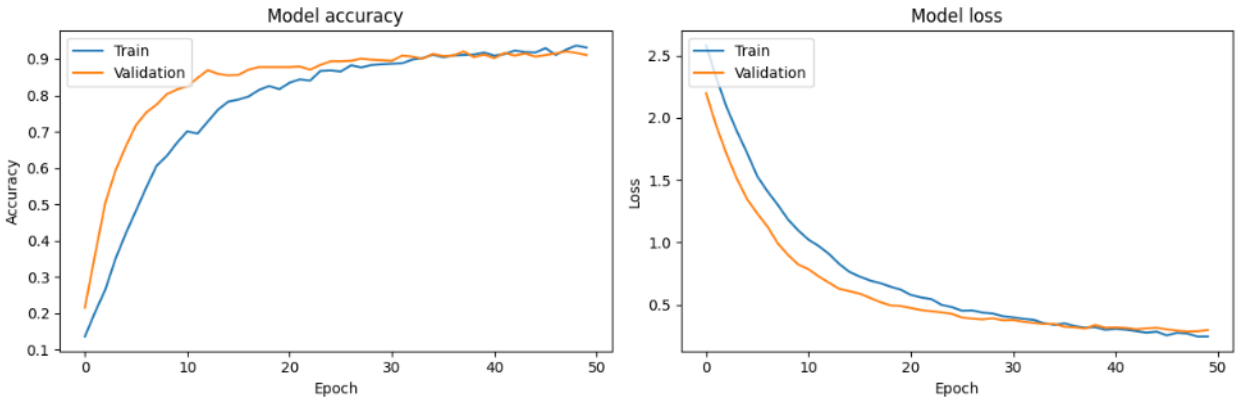

Figure 15 represents the accuracy and loss curve for experiment 4

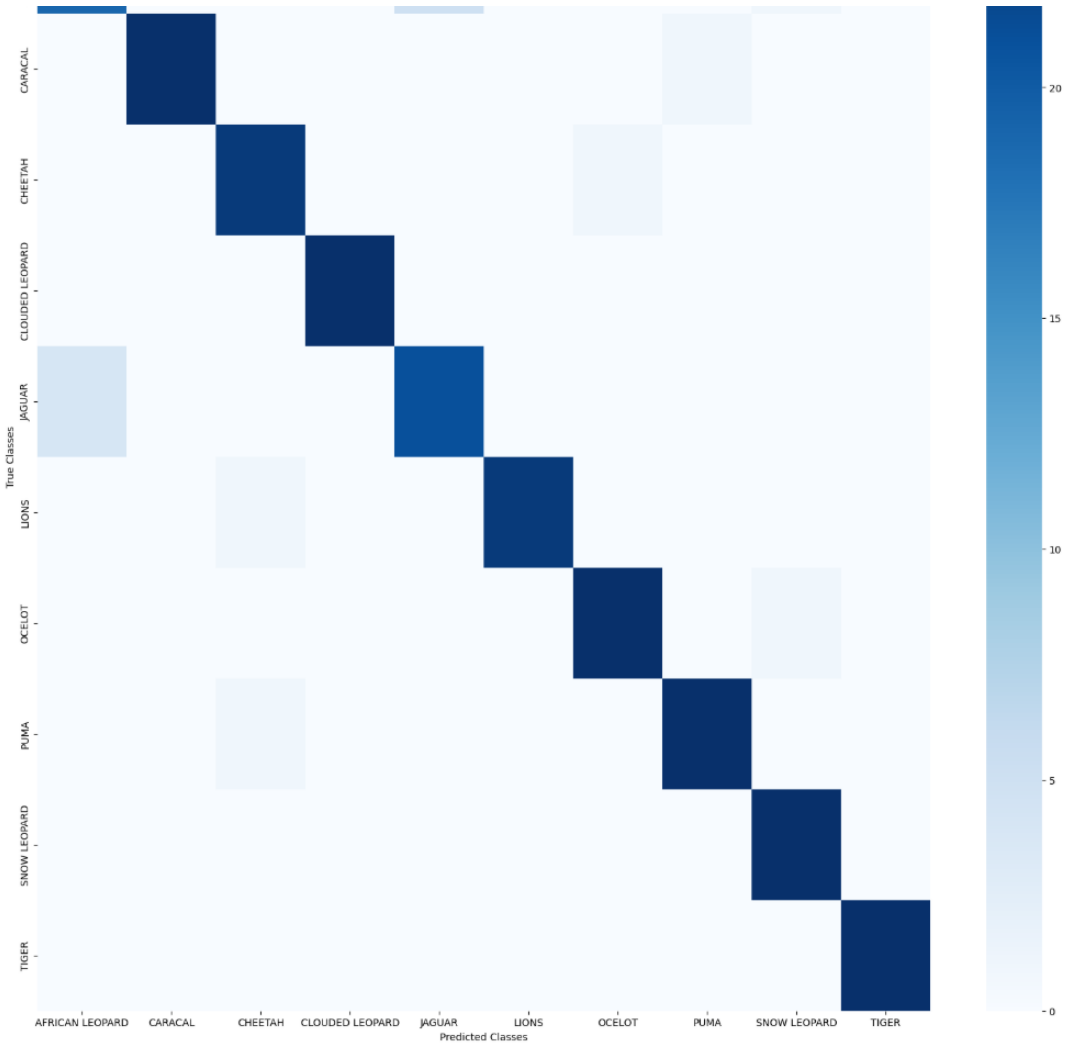

Figure 16 represents the confusion matrix for experiment 4

	precision	recall	f1-score	support
AFRICAN LEOPARD	0.83	0.76	0.79	25
CARACAL	1.00	0.96	0.98	25
CHEETAH	0.92	0.96	0.94	24
CLOUDED LEOPARD	1.00	1.00	1.00	24
JAGUAR	0.81	0.84	0.82	25
LIONS	1.00	0.96	0.98	24
OCELOT	0.96	0.96	0.96	25
PUMA	0.96	0.96	0.96	25
SNOW LEOPARD	0.92	1.00	0.96	24
TIGER	1.00	1.00	1.00	24
accuracy			0.94	245
macro avg	0.94	0.94	0.94	245
weighted avg	0.94	0.94	0.94	245

Figure 17 represents the classification report for experiment 4

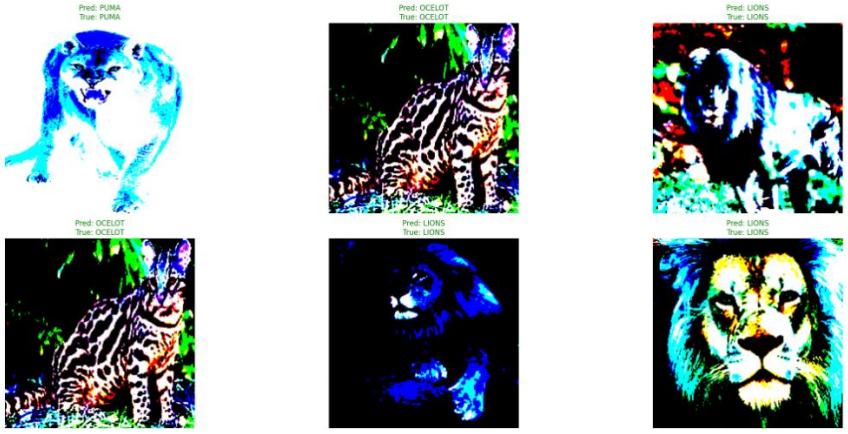

Figure 18 represents a sample of the predictions from experiment 4

Experiment 5:

For experiment 2 we split the data into a 60/30/10 ratio for the training/validation/test sets. Then we used the image data generator from keras to augment the data as stated in the data section. In this experiment all the layers were frozen except the custom layers. Adam optimizer was used as well as a learning rate of 0.0001. Early stopping, reduce learning rate on plateau was used, same as in experiment 1. 15 epochs were used to train the model. It achieved a loss of 0.2902 and an accuracy of 93.06%.

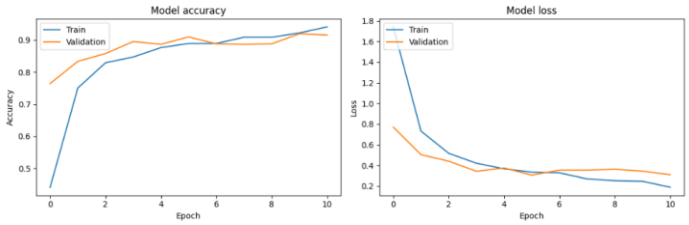

Figure 19 represents the accuracy and loss curve for experiment 5

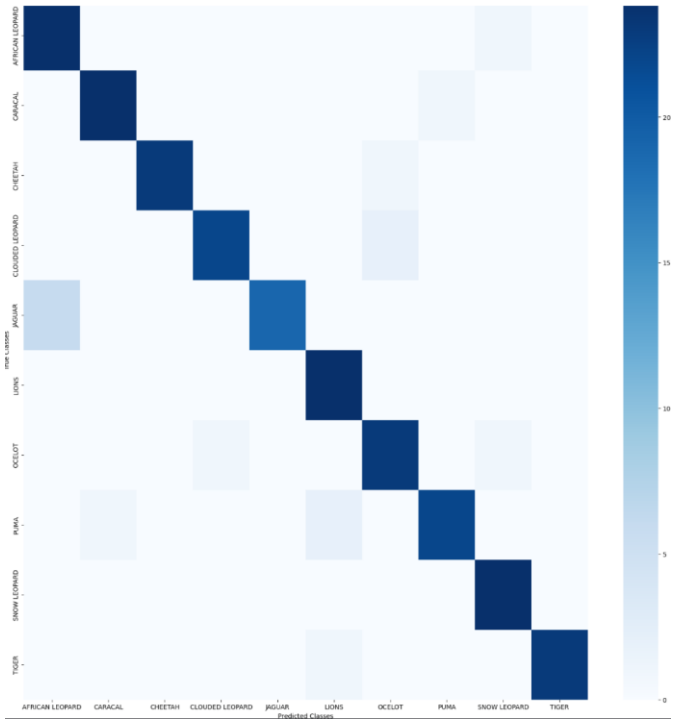

Figure 20 represents the confusion matrix for experiment 5

	precision	recall	f1-score	support
AFRICAN LEOPARD	0.80	0.96	0.87	25
CARACAL	0.96	0.96	0.96	25
CHEETAH	1.00	0.96	0.98	24
CLOUDED LEOPARD	0.96	0.92	0.94	24
JAGUAR	1.00	0.76	0.86	25
LIONS	0.89	1.00	0.94	24
OCELOT	0.88	0.92	0.90	25
PUMA	0.96	0.88	0.92	25
SNOW LEOPARD	0.92	1.00	0.96	24
TIGER	1.00	0.96	0.98	24
accuracy			0.93	245
macro avg	0.94	0.93	0.93	245
weighted avg	0.94	0.93	0.93	245

Figure 21 represents the classification report for experiment 5

Figure 22 represents a sample of the predictions from experiment 5

Timeline

The project timeline is structured as follows. In the first week, we completed data collection and preprocessing. Currently, we are in the process of training baseline models. Hyperparameter tuning is planned for the following week, followed by experiments with alternative model architectures. The final evaluation and the preparation of the project report and presentation will take place in the fifth week.

Week	Task	Status
Week 10–11	Data collection, preprocessing, and baseline model setup	Completed
Week 12	Baseline model training (VGG16, ResNet50) and initial evaluation	In Progress
Week 13	Hyperparameter tuning and optimization experiments	Planned
Week 14	Model comparison following experimentation	Planned
Week 15	Final evaluation, report writing, and presentation preparation	Planned

References

10 Big Cats of the Wild - Image Classification Dataset. *Kaggle*. Retrieved from: https://www.kaggle.com/datasets/gpiosenka/cats-in-the-wild-image-classification

Simonyan, K., & Zisserman, A. (2014). *Very Deep Convolutional Networks for Large-Scale Image Recognition (VGG)*.

He, K., Zhang, X., Ren, S., & Sun, J. (2015). *Deep Residual Learning for Image Recognition (ResNet)*.

Tan, M., & Le, Q. V. (2019). *EfficientNet: Rethinking Model Scaling for Convolutional Neural Networks*.

Szegedy, C., Vanhoucke, V., Ioffe, S., Shlens, J., & Wojna, Z. (2016). *Rethinking the Inception Architecture for Computer Vision*.

Shorten, C., & Khoshgoftaar, T. M. (2019). *A survey on image data augmentation for deep learning*.