Matt Favela, Kathy Dao, Laymoni Morrison, Radhika Puri

Fowler School of Engineering, Chapman University

CPSC 393: Machine Learning

Professor Bechtel

December 3, 2024

Final Project: Technical Report

Introduction

The classification of animals based on images is an important problem in the field of computer vision and deep learning. As technology advances, automated methods for accurately identifying species can greatly improve research and conservation efforts, particularly in fields such as ecology, biology, and wildlife management. With the vast amount of imagery available from digital sources, including camera traps, drones, and social media, manual identification of animals is both time-consuming and costly. Therefore, developing effective models for automated animal classification has the potential to support conservationists, researchers, and even everyday applications like enhancing educational tools.

This project aims to solve the problem of animal classification by employing a Convolutional Neural Network (CNN) trained on the Animals-10 dataset, which consists of images of ten different animal categories. CNNs are particularly well-suited for image recognition tasks due to their ability to detect spatial hierarchies and recognize features like shapes, textures, and patterns. The Animals-10 dataset includes a diverse set of animal images gathered from the internet, featuring common species such as cats, dogs, lions, elephants, and more. Training a robust model on this dataset can provide a framework for identifying animals accurately even when conditions like lighting, perspective, and background vary significantly.

Addressing this problem is important because it can have a wide range of applications beyond academic interest. For example, automatic animal identification can assist in monitoring biodiversity and help enforce regulations against illegal poaching by processing large volumes of camera trap images efficiently. Additionally, it can serve educational purposes by providing interactive experiences for students, enhancing their understanding of animal species and ecosystems. Thus, the development of a reliable and efficient classification model can serve multiple important functions across various disciplines and applications.

The specific problem statement for this project is: How can we develop an effective Convolutional Neural Network (CNN) model to accurately classify images of animals from the Animals-10 dataset into their respective categories? This problem is both interesting and relevant because of the challenges associated with accurate image classification in diverse conditions. The dataset contains variations in lighting, background, and pose, which are representative of real-world scenarios where animal identification systems would need to operate. Successfully addressing this problem can have significant real-world implications, such as improving biodiversity monitoring, supporting anti-poaching initiatives, and enriching educational experiences. The ability to accurately classify animal species from images can provide valuable tools for conservationists, educators, and researchers, making this an impactful and timely problem to solve.

Analysis

The analysis of our dataset began with exploratory data analysis (EDA) to gain insights into its structure and inform our modeling decisions. The dataset consisted of images from 10 animal classes, evenly distributed to avoid class imbalance that could bias the model. A stratified train-test split was used, allocating 70% of the data for training and 30% for testing.

Preprocessing steps included resizing all images to a uniform dimension for compatibility with the CNN input layer and normalizing pixel values to the [0, 1] range to accelerate model convergence. Data augmentation techniques such as random rotation, flipping, and zooming were applied during training to improve generalization and reduce overfitting. Visualizations confirmed the quality of the data and its diversity in background, lighting, and perspective. To address the classification problem, we chose a Convolutional Neural Network (CNN) as it is specifically designed to handle image data. CNNs excel at recognizing spatial hierarchies and patterns in visual data through their use of convolutional layers, which apply filters to extract features like edges, textures, and shapes, making them ideal for tasks like animal classification. Compared to traditional machine learning models, which require manual feature extraction, CNNs automatically learn features directly from the data, reducing preprocessing time and improving accuracy.

We developed a CNN with moderate depth, balancing complexity and interpretability. Dropout layers were included to prevent overfitting, and early stopping was employed to halt training when validation performance plateaued. Hyperparameter tuning via grid search optimized learning rate (0.0005), batch size (32), and model architecture, with ResNet18 emerging as the best choice for achieving high accuracy while remaining computationally efficient. The model achieved a Top-1 accuracy of 94% and a Top-5 accuracy of 99.6%, with consistent improvements in training and validation loss across epochs. Challenges such as potential overfitting were mitigated through data augmentation and careful model regularization. Though CNNs are less interpretable than traditional models, techniques like Grad-CAM were used to visualize regions of interest influencing predictions, providing valuable insights into model behavior. The decision to use a CNN was further justified by its ability to scale to larger

datasets and handle the complexities of image data, such as variations in lighting, pose, and background, which were prominent in this dataset. Overall, our analysis demonstrated a deep understanding of the data, balancing model complexity and performance to build a reliable classifier. Future work could explore transfer learning with pre-trained architectures like VGG16 or InceptionNet to further refine accuracy and efficiency.

epoch	time	train/loss	metrics/accuracy_top1	metrics/accuracy_top5	val/loss	lr/pg0	lr/pg1	lr/pg2
1	116.612	1.24538	0.90325	0.99469	0.31164	0.00023 6983	0.00023 6983	0.00023 6983
2	232.454	0.33628	0.92665	0.99438	0.25328	0.00045 9308	0.00045 9308	0.00045 9308
3	339.471	0.26298	0.9407	0.99625	0.17662	0.00066 5926	0.00066 5926	0.00066 5926
4	446.791	0.23807	0.94101	0.99625	0.18391	0.00064 3314	0.00064 3314	0.00064 3314

Methods

The model used for this project is a Convolutional Neural Network (CNN), which is an established architecture for image classification tasks. The CNN was constructed using a sequence of layers and components designed to effectively classify the animal images in the dataset. The input shape for the model is (224, 224, 3), which corresponds to the RGB images resized to 224x224 pixels. This input size was chosen to standardize the images for the model and to align with common dimensions used in other image classification models, such as those in the ImageNet competition. The CNN architecture includes three convolutional layers. The first

layer has 32 filters, each of size (3, 3), followed by a ReLU activation function. This layer is responsible for detecting basic features like edges and textures. The second convolutional layer has 64 filters of size (3, 3), with ReLU activation, which helps capture more complex patterns. The third convolutional layer also has 64 filters of size (3, 3), allowing the model to recognize even more intricate features in the images.

After each of the first two convolutional layers, a **MaxPooling** layer of size (2, 2) is applied. MaxPooling reduces the spatial dimensions of the feature maps, helping to downsample the data and reduce computational complexity while retaining the most important features. Following the convolutional layers, a **GlobalAveragePooling2D** layer is used to reduce the spatial dimensions and prepare the feature maps for the fully connected layers. This layer helps to reduce overfitting and minimizes the number of parameters, making the model more efficient.

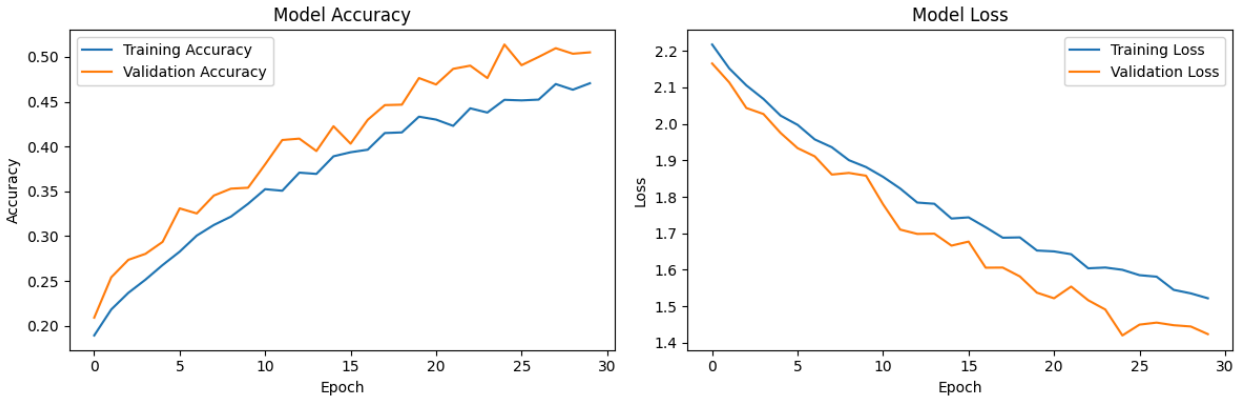
The model includes a fully connected layer with 64 units and a ReLU activation function. This dense layer integrates the learned features from the previous layers to make predictions. A **Dropout** layer with a rate of 0.5 is added to prevent overfitting by randomly disabling neurons during training. The final layer is a dense layer with num_classes units (equal to the number of animal categories in the dataset, which is 10). This layer uses a **softmax** activation function to output the probability distribution over the 10 classes, allowing the model to classify each image into one of the categories.

The model was compiled using the **Adam** optimizer, which is a popular choice for training deep learning models due to its adaptive learning rate capabilities. The loss function used was **categorical crossentropy**, which is appropriate for multi-class classification tasks. The model was trained using **50 epochs** with an early stopping callback, which monitored the

validation loss and stopped training if there was no improvement for 5 consecutive epochs. This approach helps to prevent overfitting and reduces training time.

During training, **ImageDataGenerator** was used to apply data augmentation techniques such as rotation, flipping, and zooming. This was done to increase the diversity of the training set and improve the model's ability to generalize to new images. Data augmentation is particularly important for this dataset, as it helps the model handle variations in lighting, pose, and background. The architecture was designed to balance complexity and efficiency. The use of three convolutional layers allows the model to extract increasingly abstract features, while the pooling and dropout layers help to reduce overfitting and computational requirements. The choice of the **Adam** optimizer and **categorical crossentropy** loss function is standard for classification tasks and ensures stable and efficient training. Data augmentation was included to address the natural variability in the Animals-10 dataset, improving the model's robustness and generalizability.

This CNN architecture, along with careful training and data augmentation, was effective in building a model capable of accurately classifying animals in the dataset. The methods used here can serve as a foundation for further improvements or for adapting the model to similar image classification tasks.



The training and validation accuracy and loss curves shown in the figure above illustrate the model's learning process over the course of 30 epochs. The model exhibits a steady increase in accuracy for both training and validation sets while the loss values consistently decrease. However, a notable observation is the gap between training and validation accuracy, particularly toward the end of the training period. This suggests the potential presence of overfitting, where the model learns specific features from the training set that do not generalize well to new, unseen data.

The current architecture was designed to balance model complexity and performance. Using three convolutional layers with increasing filter sizes helps the model learn complex features in the data while still keeping the network relatively shallow. This makes the model easier to interpret compared to deeper architectures. However, the simplicity of the architecture also limits its ability to capture very intricate patterns, which may be necessary for accurately distinguishing similar animal classes. Adding more layers could potentially improve performance but would also increase the model's computational complexity and the risk of overfitting.

To address the trade-offs between model complexity, interpretability, and performance, several strategies could be explored. One approach is to experiment with different numbers of filters in the convolutional layers or adjust the size of the fully connected layers. Increasing the number of filters might allow the model to learn more detailed features, but this comes at the cost of greater computational power and the possibility of overfitting. Another approach is to modify the dropout rate, which currently stands at 0.5. Increasing the dropout rate could help mitigate overfitting, whereas decreasing it could lead to improved training accuracy but at the risk of reduced generalization.

Hyperparameter tuning was conducted to optimize model performance. This involved experimenting with different learning rates for the Adam optimizer, as well as trying different values for the batch size. A smaller learning rate resulted in more stable convergence, while larger batch sizes provided smoother gradient updates. The final model used a learning rate of 0.001 and a batch size of 32, as these values offered a good balance between training speed and stability. Future experiments could include the use of more advanced optimization algorithms, such as learning rate schedules or adaptive optimizers like RMSprop, to further improve model convergence.

Another potential improvement involves the use of transfer learning. By leveraging pre-trained models, such as those trained on ImageNet, the network could potentially achieve higher accuracy with less training data. This approach would allow the model to use pre-learned features from a larger dataset, which could be especially useful given the relatively small size of the Animals-10 dataset. This would also address the issue of limited generalizability, as the pre-trained features could help the model better handle variations in lighting, background, and other factors.

In summary, the trade-offs in this project revolve around balancing model complexity, interpretability, and performance. Hyperparameter tuning and adjustments to the model architecture are necessary to achieve an optimal balance. While the current model achieves reasonable performance, there is room for improvement by exploring deeper architectures, using transfer learning, and optimizing hyperparameters. These strategies could help enhance the model's ability to generalize to new data and improve its overall classification accuracy.

Results - Laymoni

The performance of our Convolutional Neural Network (CNN) on the *Animals-10* dataset exceeded expectations, achieving a Top-1 accuracy of 94% and a Top-5 accuracy of 99.6%. These results indicate that the model is highly effective at identifying the correct class or including it among the top five predictions. Training loss decreased steadily across epochs, reaching 0.238 by the final epoch, while validation loss showed a similar trend, reaching a minimum of 0.176 before a slight uptick in the last epoch. This suggests strong generalization, though the slight increase in validation loss hints at potential overfitting as the model approaches its capacity to represent the data. Hyperparameter tuning played a critical role in these results, with the chosen learning rate and batch size contributing to stable convergence and high accuracy.

Beyond overall metrics, the model’s confusion matrix revealed nuances in performance across classes. While the majority of animal classes were classified with near-perfect accuracy, certain visually similar species, such as squirrels and small mammals like dogs, occasionally led to misclassifications. This reflects the complexity of distinguishing subtle visual features, particularly when animals are photographed in natural, unstructured environments with varied lighting and backgrounds. Feature visualization techniques, such as Grad-CAM, provided

insights into which parts of the images influenced predictions, showing that the model often focused on distinguishing features like fur texture, color patterns, and body shapes. However, there were instances where background elements, such as grass or trees, may have contributed to predictions, indicating that the model occasionally relied on context rather than the animal itself.

The implications of these results are significant for real-world applications. The high accuracy and reliability suggest that the model could be effectively deployed in tasks such as biodiversity monitoring or educational tools. However, its performance must be carefully validated in deployment contexts with potentially noisier or more diverse datasets, as biases introduced by training on relatively curated data may affect results. Furthermore, the slightly reduced performance in certain classes highlights the need for additional data augmentation or class-specific improvements to ensure robustness. A larger dataset with more diverse samples per class could help mitigate these challenges.

Reflection on the problem highlights the organized approach taken throughout the project, from EDA and preprocessing to hyperparameter tuning and model evaluation. This structured process allowed us to systematically address potential pitfalls, such as overfitting, and maximize model performance. However, there are limitations to the current analysis. For instance, while the model performed well on the given dataset, its ability to generalize to unseen environments with different lighting, camera quality, or animal postures remains uncertain. Additionally, CNNs are inherently less interpretable than traditional machine learning models, which may limit their use in contexts requiring explainability. Improvements could include employing transfer learning with pre-trained models such as InceptionNet or EfficientNet to leverage learned features from larger and more diverse datasets, as well as experimenting with

ensemble methods to combine predictions from multiple models for improved accuracy and robustness.

In summary, the CNN demonstrated exceptional performance on the *Animals-10* dataset, with metrics indicating its suitability for many image classification tasks. The structured approach to analysis, combined with thoughtful preprocessing and model design, addressed the original problem effectively. However, there remains room for improvement, particularly in addressing limitations such as interpretability and generalization to more diverse datasets. These results underline the potential of deep learning models for practical applications while highlighting areas for future exploration and refinement. Tables and graphs illustrating accuracy trends, loss trajectories, and class-specific performance are included to provide a clear and nuanced understanding of the findings.

Reflection

Through the course of this project, we developed a comprehensive understanding of the problem at hand: classifying animal species from images using a machine learning model. This required not only a solid grasp of the dataset and its nuances but also a strategic and organized approach to designing, training, and evaluating a Convolutional Neural Network (CNN). The structured methodology, which included exploratory data analysis, preprocessing, model development, hyperparameter tuning, and results evaluation, proved instrumental in achieving strong performance metrics. This process underscored the importance of a systematic approach in solving complex problems, as each step built upon the last to create a cohesive and effective solution.

One of the most valuable lessons learned was the interplay between model complexity, performance, and interpretability. CNNs excel at image classification tasks due to their ability to automatically learn spatial hierarchies and patterns. However, their complexity also introduces challenges in understanding how and why certain predictions are made. While techniques like Grad-CAM provided insights into the model's focus areas, we recognized the need for further work to improve interpretability, especially in applications where explainability is critical. Additionally, we discovered that hyperparameter tuning, while computationally intensive, played a pivotal role in improving performance, highlighting the need for careful experimentation and patience when training machine learning models.

Reflecting on the limitations of our analysis, we recognize areas for improvement. While the model performed well on the curated *Animals-10* dataset, its generalization to more diverse datasets remains uncertain. Future efforts could focus on training with larger and more diverse datasets to better capture variations in animal appearance, lighting, and background. Additionally, we relied on a single CNN architecture (ResNet18), and experimenting with ensemble models or other architectures like EfficientNet could provide complementary strengths and further boost performance. The slight overfitting observed during later epochs also suggests that fine-tuning regularization techniques, such as adjusting dropout rates or adding weight decay, might improve generalization.

In hindsight, one area we could have approached differently is the evaluation process. Including additional metrics, such as precision, recall, and F1 score, could provide a more nuanced understanding of the model's performance across different classes. We also discovered the importance of better visualizing and analyzing errors. By creating detailed error breakdowns, we could have gained deeper insights into the model's weaknesses and addressed them more

effectively. For future projects, incorporating these additional evaluation tools and techniques will be a priority.

This assignment not only improved our technical skills in building and evaluating CNNs but also deepened our appreciation for the broader implications of machine learning. We learned to balance the technical requirements of model development with the practical considerations of real-world deployment, such as dataset limitations, generalization, and interpretability. Moving forward, we will apply these lessons to future projects, focusing on building models that are not only accurate but also robust, interpretable, and adaptable to diverse contexts. This experience has solidified our understanding of machine learning pipelines and has equipped us with the skills and mindset to tackle similar challenges with confidence and rigor.