**Animal detection in the wild using Computer Vision and AI**

Akom, Jerry

(Computer vision: object detection, classification, segmentation, tracking)

Image classification is a core task in computer vision (Computer Vision, SOTA) that involves assigning labels to images based on their visual content (Wang and Su. 2019). With the advent of deep learning, Convolutional Neural Networks (CNNs) have become the dominant approach for such tasks, offering superior performance by learning hierarchical feature representations directly from pixel data. This project utilizes the Animal Image Dataset, AID (Banerjee; 2022), comprising over 5,000 images across 90 animal classes, to explore the capabilities of modern CNN architectures in classifying a diverse range of animal species.

**2. Background and Related Work**

Before the invention of CNNs, image classification approaches relied on hand-crafted features combined with classical machine leering classifiers like SVM and Radom Forests, but these have been largely replaced by CNNs, which learn spatial hierarchies of features directly from raw image pixels. Several CNN architectures have emerged as benchmarks (https://paperswithcode.com/task/image-classification) in the field, each offering a unique balance of performance and efficiency.

ResNet50, one of the most influential deep learning architectures, was introduced by He et al. (2015), is a deep CNN with 50 layers. Its residual connections, which allow the network to learn identity mappings and mitigate the vanishing gradient problem, a major issue in earlier deep CNNs (He et al. 2015). These skip connections effectively allow gradients to flow backward through the network more easily, enabling the successful training of very deep models. It uses bottleneck blocks to reduce computational cost while maintaining representational power (He et al. 2015).

MobileNetV2, introduced by Sandler et al. (2018), was designed with efficiency in mind, specifically targeting mobile and embedded vision applications. It builds upon its predecessor by incorporating two key architectural innovations: inverted residual blocks and linear bottlenecks. Unlike traditional CNNs, which typically increase the number of filters through layers, MobileNetV2 first expands the number of channels, applies lightweight depth wise separable convolutions, and then projects the features back to a lower-dimensional space. This structure dramatically reduces the number of trainable parameters and floating-point operations (FLOPs), making the network highly efficient without significantly sacrificing accuracy. The inverted residual connections also improve gradient flow, allowing the network to train deeper representations with fewer computational resources than standard CNNs.

EfficientNet by Tan and Le (2019), introduced a novel compound scaling method that jointly optimizes the network's depth, width, and input resolution. EfficientNetB3 offers a good trade-off between accuracy and computational efficiency (Tan and Le 2019). The architecture is built around MBConv blocks (Mobile Inverted Bottleneck Convolution), which combine the efficiency of MobileNetV2 with squeeze-and-excitation (SE) blocks that improve channel-wise feature recalibration. By systematically scaling all three dimensions of the network using fixed coefficients, EfficientNetB3 achieves state-of-the-art accuracy with significantly fewer parameters than comparable networks (Tan and Le 2019).

The development of large-scale datasets such as ImageNet (Deng et al. 2009) and CIFAR-10 (Krizhevsky, 2009) has enabled the training of high-capacity CNNs that generalize well across diverse image domains. Researchers have been able to effectively utilize these pretrained models to address classification tasks with limited training data through the process of transfer learning. Transfer learning is a technique where a model trained on a large, general-purpose dataset is fine-tuned on a smaller, domain-specific dataset. It allows models to reuse learned visual features and adapt to new classification problems efficiently.

To enhance the robustness and generalization of deep learning models, **image augmentation techniques** have become a fundamental part of the training pipeline in image classification tasks. These methods involve applying a range of label-preserving transformations—such as rotation, flipping, scaling, cropping, brightness adjustment, and noise injection—to generate diverse variants of existing training images. This synthetic diversity effectively enlarges the dataset, helping models learn invariant features and reducing overfitting, especially when working with limited or imbalanced data (Shorten & Khoshgoftaar, 2019; Krizhevsky et al., 2012).

Several works have explored deep learning for fine-grained image classification, including animal species recognition. Datasets like ImageNet and CIFAR-10 contain some animal images and have been used to benchmark classification models on wildlife categories (Wang and Su. 2019). However, few works have focused specifically on multi-class animal classification with as many categories as in the AID used in this study. Even looking at the Animal Image Classification Dataset (AICD. 2025) and other Animal Kingdom datasets (AKD, 2025) show that it is a domain that is hardly explored.

This work builds upon these foundations by conducting a comparative analysis of three state-of-the-art CNN models in the context of wild animal classification. It applies image augmentation techniques and transfer learning by fine-tuning these pretrained models on the AID to evaluate their classification performance. Metrics such as accuracy, precision, recall, and parameter count are used to compare the models. The objective is to assess the trade-offs between model complexity, computational efficiency, and predictive accuracy, thereby informing the choice of architecture for image classification tasks involving fine-grained, multi-class datasets.

**2. Methodology**

This study follows a structured approach for Transfer learning & fine-tuning as proposed in by Chollet (2020, 2023). The methodology encompasses dataset preparation, model selection, transfer learning, training configuration, and evaluation using key performance metrics.

**2.1 Dataset Preparation**

The Animal Image Dataset from Kaggle contains approximately 5,400 images across 90 animal categories, with roughly 60 images per class. The dataset was first cleaned by resizing all images to match the input size required by each model (224×224 for MobileNetV2 and ResNet50; 300×300 for EfficientNetB3). The data was then split into training (70%), validation (15%), and test (15%) sets. Images were normalized to a pixel value range of [0, 1], and appropriate preprocessing functions were applied for each model’s expected input format. Data augmentation techniques such as flips, rotations, shifts and zooms were used on the training set to reduce overfitting and enhance model generalization on a second run.

**2.2 Model Selection and Transfer Learning**

Three pretrained CNN models from Keras namely; MobileNetV2, EfficientNetB3, and ResNet50, were used as base architectures with ImageNet weights. The original top classification layers were removed and replaced with a new fully connected head consisting of a Global Average Pooling layer, and a final Softmax output layer of size 90 (matching the number of classes). During training, the base model layers were initially frozen to train only the new classification head. Fine-tuning was then performed by unfreezing the top layers of the base model to adapt the learned features to the new domain.

**2.3 Training and Evaluation**

All models were trained using the Adam optimizer with a learning rate of e-4 and categorical cross-entropy loss. Training was performed for a maximum of 10 epochs with early stopping based on validation loss to prevent overfitting. A batch size of 32 was used across all models for consistency. Model performance was evaluated using accuracy, precision, recall, F1-score and confusion matrices. The number of trainable parameters was also recorded as a measure of model complexity. All experiments were conducted using TensorFlow using Google Colab + T4 GPU system (12.7 GB RAM, 15 GB GPU, 112.5 GB Disk).

**3. Results**

Results from the three models, MobileNetV2, EfficientNetB3, and ResNet50, and two datasets, only normalization and normalization + augmentation are presented in this section. All results are derived from the held-out test dataset (15% of total).

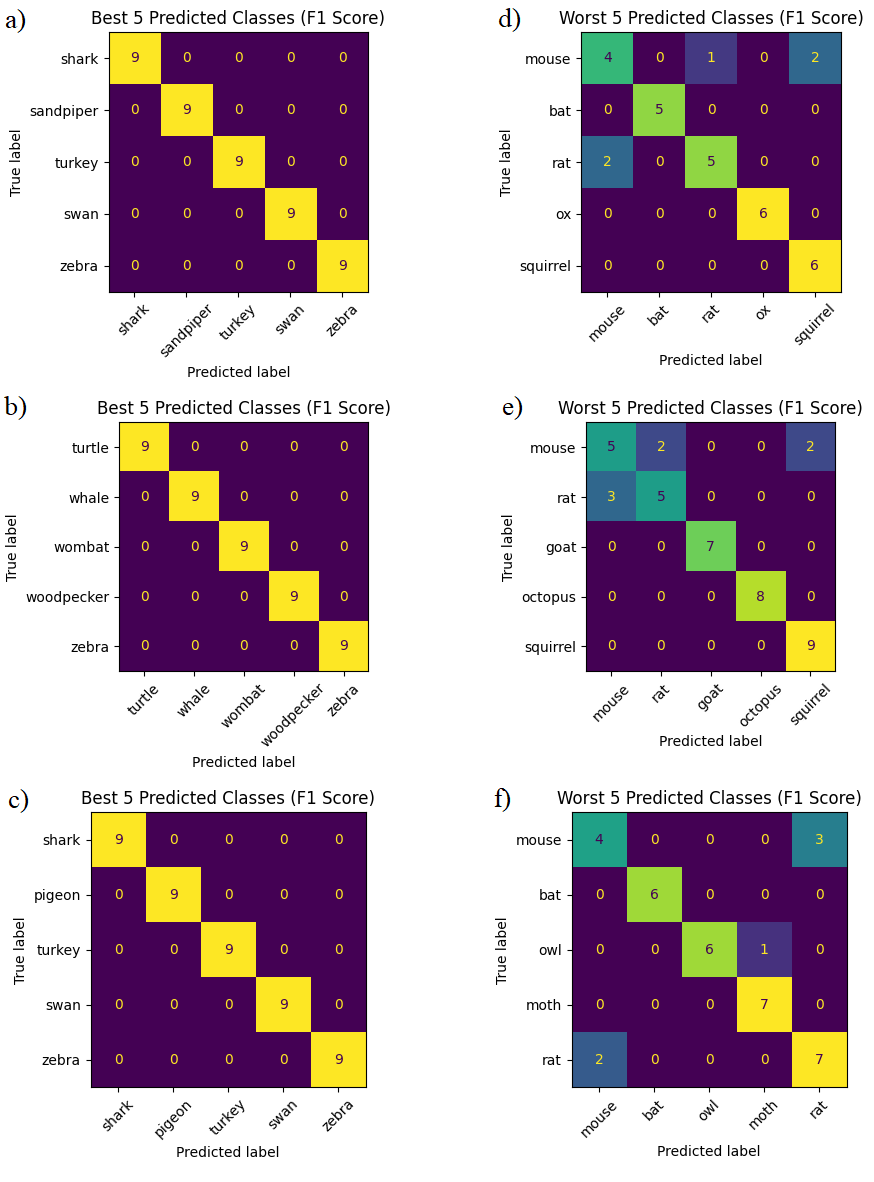


Figure 1. Best five predictions for MobileNetV2 (a), EfficientNetB3 (b), ResNet50 (c) and Worst five predictions for MobileNetV2 (d), EfficientNetB3 (e), ResNet50 (f)

Table 1: Classification Metrics on Test Set

|  |  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- | --- |
| **Model** | **Augmentation** | **Accuracy (%)** | **Precision (%)** | **Recall (%)** | **F1-Score (%)** | **Parameters (M)** | **Misclassifications** |
| MobileNetV2 | No | 88 | 89 | 88 | 88 | ~6.6 | 98 |
| EfficientNetB3 | No | 95 | 95 | 95 | 95 | ~11.1 | 40 |
| ResNet50 | No | 90 | 91 | 90 | 90 | ~24.1 | 79 |
| MobileNetV2 | Yes | 88 | 89 | 88 | 88 | ~6.6 | 96 |
| EfficientNetB3 | Yes | 95 | 95 | 95 | 95 | ~11.1 | 41 |
| ResNet50 | Yes | 90 | 91 | 90 | 90 | ~24.1 | 84 |

**3.1 Classification Performance**

Table 1 summarizes the classification accuracy, precision, recall, F1-score, model parameters and overall misclassificationss for each model and dataset with and without augmentation. EfficientNetB3 achieved the highest overall accuracy (95%), while MobileNetV2 and ResNet50 had almost similar performance despite having about 3.5 times fewer parameters. Image augmentation did not lead to any improvements in overall accuracy for all three models. Since the data augmentations did not provide any improvements in model performance, only the results are not presented further.

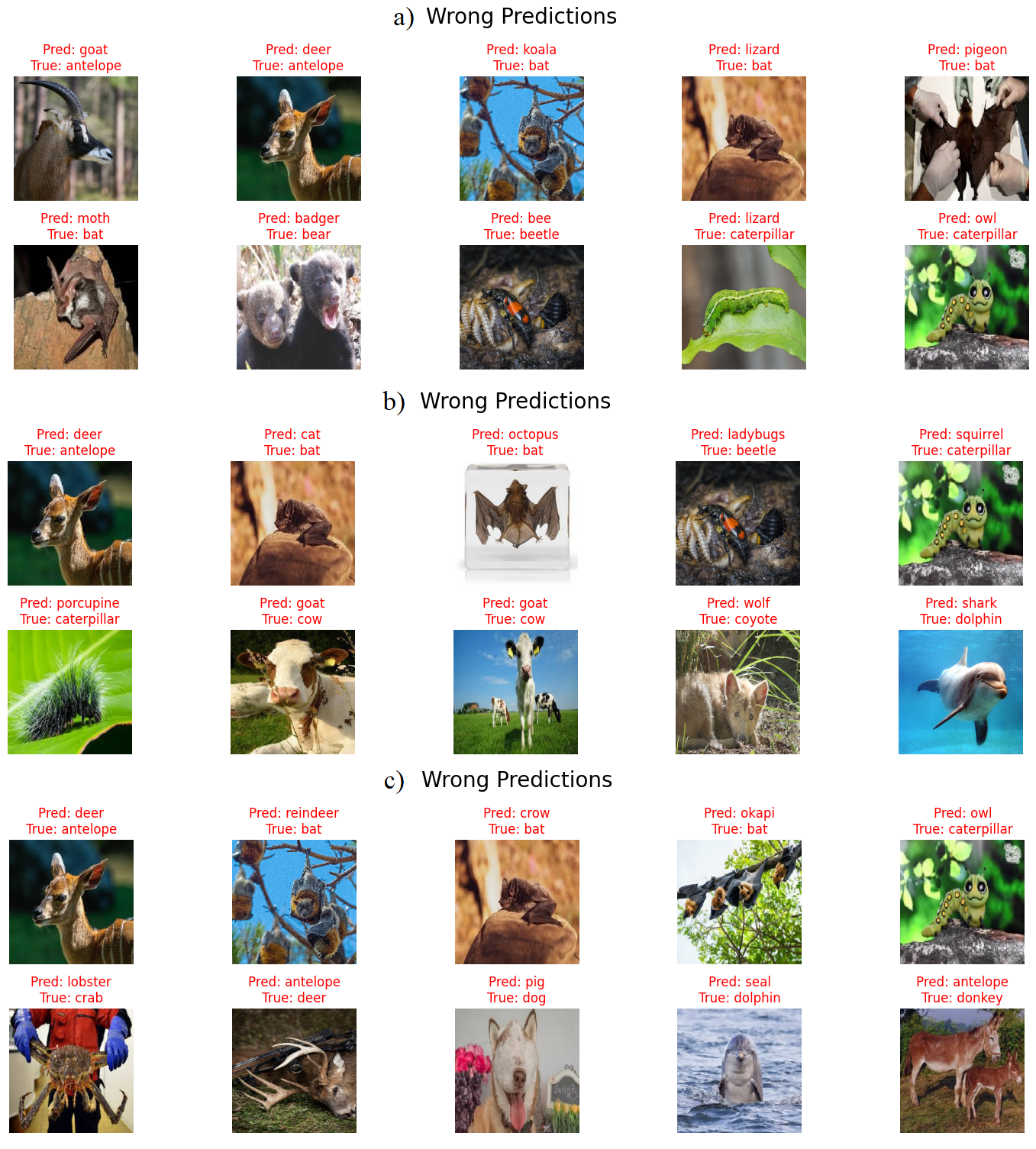


Figure 2. Some wrong predictions for MobileNetV2 (a), EfficientNetB3 (b), ResNet50 (c)

**3.2 Confusion Matrix and Visual Analysis**

Confusion matrices were generated to assess model performance. However, due to the large number of classes (90), confusion matrix could not be created to include all classes. The best and worst five classes were calculated for each model based on their F1-scores (Figure 1). Sample misclassified images and model predictions are also illustrated in Figure 2.

While all models showed high accuracy on commonly represented classes, some misclassifications were observed among visually similar species. For example: rats, mouse and squirrel; goats, deer, antelopes and cows; bats and birds were misclassified for each other. EfficientNetB3 demonstrated better separation across fine-grained categories, likely due to its deeper scaling and squeeze-excitation blocks (Sandler et al. 2018).

**4. Discussion**

The comparative evaluation of MobileNetV2, EfficientNetB3, and ResNet50 on the Animal Image Dataset highlights key trade-offs between accuracy, computational efficiency, and model complexity. EfficientNetB3 consistently delivered the best classification performance across all evaluated metrics, demonstrating its strength in balancing depth, width, and resolution via compound scaling. This model excelled particularly in distinguishing fine-grained animal classes, likely due to its inclusion of MBConv and squeeze-and-excitation blocks that enhance feature representation (Sandler et al. 2018).

ResNet50, was behind EfficientNetB3 in overall accuracy, still provided strong results, particularly in recall. Its deep residual architecture allowed it to learn robust features, especially for complex and ambiguous classes (He et al. 2015): However, its relatively large parameter count could increase processing and limit its applicability in low-resource or real-time scenarios.

In contrast, MobileNetV2 offered the smallest model footprint, making it ideal for resource limited systems like mobile devices. Its accuracy was not significantly lower than ResNet50, hence it remains a viable option where latency or memory constraints are critical factors (Tan and Le 2019). These results underscore that model selection should be driven by task-specific constraints: accuracy for high-stakes classification, or speed and size for lightweight deployment.

**5. Conclusion**

This study presented a transfer learning-based image classification project using three prominent CNN architectures on a multi-class animal dataset. By analysing their performance in terms of accuracy and model parameters, it demonstrated how modern CNNs can be effectively applied to real-world classification tasks. The findings suggest that EfficientNetB3 offers the best balance of accuracy and efficiency, ResNet50 remains a strong baseline with deep representation power, and MobileNetV2 is best suited for fast, resource-constrained applications. In this project, image augmentation did not improve overall accuracy on the Animal Image Dataset. At the end of the day, it did not ultimately contribute to more reliable and generalizable classification as expected.

Future work could explore other CNN architectures to see which one could produce accuracies better than 95% if that is required. Additionally, expanding the dataset or applying these models to other fine-grained classification problems could yield further insights into generalization capabilities.

**References**

AICD. (2025). *Animal Image Classification Dataset.* <https://paperswithcode.com/dataset/animal-image-classification-dataset> (Last Accessed: 05/11/2025).

AKD. (2025). *Animal Kingdom Datasets.* <https://paperswithcode.com/datasets?q=Animal+Kingdom&v=lst&o=match> (Last Accessed: 05/11/2025).

Banerjee, S. (2022). *Animal Image Dataset (90 Different Animals).* <https://www.kaggle.com/datasets/iamsouravbanerjee/animal-image-dataset-90-different-animals/data> (Last Accessed: 05/11/2025).

Chollet, F. (2020, 2023). *Complete guide to transfer learning & fine-tuning in Keras.* Transfer learning & fine-tuning. <https://keras.io/guides/transfer_learning/> (Last Accessed: 05/11/2025). Bottom of Form

Computer Vision, SOTA. <https://paperswithcode.com/area/computer-vision> (Last Accessed: 05/11/2025).

Deng, J., Dong, W., Socher, R., Li, L.-J., Li, K., & Fei-Fei, L. (2009). *Imagenet: A large-scale hierarchical image database*. In 2009 IEEE conference on computer vision and pattern recognition (pp. 248–255). <https://doi.org/10.1109/CVPR.2009.5206848>

He, K., Zhang, X., Ren, S., & Sun, J. (2016). *Deep Residual Learning for Image Recognition*. In Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition (CVPR), 770–778. <https://doi.org/10.1109/CVPR.2016.90>

Krizhevsky, A. (2009). *Learning Multiple Layers of Features from Tiny Images.* <https://www.cs.toronto.edu/~kriz/learning-features-2009-TR.pdf> (Last Accessed: 05/11/2025).

Krizhevsky, A., Sutskever, I., & Hinton, G. E. (2012). *ImageNet classification with deep convolutional neural networks*. In Advances in neural information processing systems (Vol. 25). <https://papers.nips.cc/paper_files/paper/2012/file/c399862d3b9d6b76c8436e924a68c45b-Paper.pdf>

Sandler, M., Howard, A., Zhu, M., Zhmoginov, A., & Chen, L. C. (2018). *MobileNetV2: Inverted Residuals and Linear Bottlenecks*. In Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition (CVPR), 4510–4520. <https://doi.org/10.1109/CVPR.2018.00474>

Shorten, C., Khoshgoftaar, T.M. (2019). *A survey on Image Data Augmentation for Deep Learning*. J Big Data 6 (60) :<https://doi.org/10.1186/s40537-019-0197-0>

Tan, M., & Le, Q. V. (2019). *EfficientNet: Rethinking Model Scaling for Convolutional Neural Networks*. In Proceedings of the 36th International Conference on Machine Learning (ICML), 6105–6114. <https://arxiv.org/abs/1905.11946>

Wang, S., & Su, Z. (2019). *Metamorphic Testing for Object Detection Systems*. <https://arxiv.org/abs/1912.12162>