Master of Science in Computer Science
June 2024

Evaluation of Lightweight CNN Architectures for Multi-Species Animal Image Classification

Siddharth Reddy Aleti

Karthik Kurakula

Faculty of Computing, Blekinge Institute of Technology, 371 79 Karlskrona, Sweden

This thesis is submitted to the Faculty of Computing at Blekinge Institute of Technology in partial fulfilment of the requirements for the degree of Master of Science in Computer Science. The thesis is equivalent to 20 weeks of full time studies.

The authors declare that they are the sole authors of this thesis and that they have not used any sources other than those listed in the bibliography and identified as references. They further declare that they have not submitted this thesis at any other institution to obtain a degree.

Contact Information:
Author(s):
Siddharth Reddy Aleti
E-mail: sial21@student.bth.se

Karthik Kurakula
E-mail: kaku21@student.bth.se

University advisor:
Dr. Prashant Goswami
Department of Computer Science

Faculty of Computing
Blekinge Institute of Technology
SE–371 79 Karlskrona, Sweden

Internet : www.bth.se
Phone : +46 455 38 50 00
Fax : +46 455 38 50 57

Abstract

Background. Recently, animal image classification has involved the impracticality of deep learning models due to high computational demands, limiting their use in wildlife monitoring. This calls for the crucial necessity of lightweight models for real-time animal identification in resource-limited environments like wildlife cameras and mobile devices. Achieving accurate animal classification in these settings becomes a crucial concern for advancing classification methods.

Objectives. The goal of this research is to evaluate lightweight transfer learning models for classifying animal images while balancing computational efficiency and accuracy. The objectives include analyzing the models' performance and providing model selection criteria based on performance and efficiency for resource-constraint environments. This study contributes to the advancement of machine learning in wildlife preservation and environmental monitoring, which is critical for accurate species identification.

Methods. The proposed methodology involves conducting a thorough literature review to identify lightweight transfer learning models for image classification. The Animal-90 dataset was utilized, comprising images of ninety distinct animal species. To train the dataset, selected pre-trained models, MobileNetV2, EfficientNetB3, ShuffleNet, SqueezeNet, and MnasNet were employed with custom classification heads. A 5-fold Cross-Validation technique was used to validate the model. A combined metric approach is applied to rank the models based on the results from the metrics, Accuracy, Inference time, and number of parameters.

Results. The experimental outcomes revealed EfficientNetB3 to be the most accurate but also at the same time it has the highest number of parameters among other models. Friedman's test has rejected the null hypothesis of models having similar performance. The combined metric approach ranked ShuffletNet as the top model among the compared models in terms of performance and efficiency.

Conclusions. The research unveiled the commendable performance of all the models in animal image classification, with ShuffleNet achieving the top rank among all other models in terms of accuracy and efficiency. These lightweight models, especially ShuffleNet, show promise in managing limited resources while ensuring accurate animal classification and confirming their reliability in wildlife conservation.

Keywords: *Lightweight architecture, Animal Image Classification, ShuffleNet Performance, Resource-Limited Environments.*

Acknowledgments

We are deeply grateful to Dr. Prashant Goswami, our thesis supervisor, for providing us with the opportunity to work under his guidance. His continuous support, invaluable advice, and insightful feedback were instrumental in overcoming the challenges we faced throughout our research. Without his encouragement and expert guidance, the completion of this thesis would not have been possible. We feel exceptionally fortunate to have had such an outstanding mentor.

We would also like to express our heartfelt appreciation to our parents, and family for their unwavering support and encouragement. Their belief in us and constant motivation have been a source of strength and inspiration throughout this journey. We are grateful to our friends for their understanding, support, and companionship. Their patience and encouragement have been invaluable, making this challenging journey more manageable and enjoyable.

Contents

List of Figures

List of Tables

Chapter 1

Introduction

In the expansive field of machine learning, computers now possess the remarkable ability to extract insights and make informed decisions from data without explicit programming. Although machine learning holds great promise, developing effective models is difficult, especially in fields like computer vision where large amounts of diverse data are necessary [9]. Transfer learning, an effective approach in recent years, addresses this challenge by leveraging knowledge from one task to enhance the performance of related tasks, particularly in image classification [10, 11]. This strategic approach has demonstrated its effectiveness and firmly established itself as a cornerstone in real-world machine-learning applications.

Although transfer learning has shown great promise, the application of complex models in resource-constrained settings—like those found in ecological research and wildlife monitoring—requires careful consideration of computational efficiency. This emphasizes how crucial lightweight transfer learning models are: they should be designed to need less computing power without sacrificing prediction accuracy [12]. As the demand for practical applications, like animal image classification in the field of wildlife conservation, continues to grow, the pursuit of lightweight models becomes essential. In this landscape, EfficientNetB3, MobileNetV2, ShuffleNet, SqueezeNet, and MnasNet emerge as pivotal convolutional neural network architectures. Each of these models brings unique advantages:

- **EfficientNetB3** is part of the broader EfficientNet series [13], which strikes a balance between computational efficiency and accuracy, making it adept at image classification, object detection, and segmentation.
- **MobileNetV2** [14] is tailored for mobile and embedded vision applications, enhancing calculation speed without compromising understanding, and proving valuable in resource-constrained environments.
- **ShuffleNet** [6] utilizes pointwise group convolution and channel shuffle, significantly reducing computational cost while maintaining competitive performance.
- **SqueezeNet** [7] offers AlexNet-level accuracy with far fewer parameters, exemplifying how architectural innovations can achieve efficient, compact models.
- **MnasNet** [8] employs a platform-aware neural architecture search approach that optimizes latency on mobile devices, blending efficiency with robustness.

Their adaptability through transfer learning positions these models as crucial players in real-world deployments, offering efficiency and accuracy in diverse image-processing tasks within the realm of wildlife conservation and ecological research.

Understanding and recognizing the wide variety of creatures in the vast realm of wildlife is essential for ecological management and conservation. This is where image classification steps in [15], specifically focusing on a dataset with 90 different animal classifications. As opposed to conventional methods that might focus on a particular species, our objective is to train the model to identify a wide range of animals, including mammals, insects, birds, and reptiles. This broad perspective allows us to capture the complex structure of life, offering a richer understanding of biodiversity. The challenge lies in the sheer diversity of 90 classifications, requiring sophisticated deep learning techniques to accurately classify this wide spectrum of wildlife. It's an exploration into the heart of biodiversity, utilizing advanced technology to unlock insights that can shape our approach to wildlife preservation.

The primary objective of this thesis is to contribute to the advancement of image classification techniques in the specific context of animal identification through the exploration and evaluation of lightweight transfer learning models [16]. The overarching goal is to bridge the gap between the demand for accurate species classification in wildlife monitoring scenarios and the constraints imposed by limited computational resources. Through a thorough examination of existing literature, the selection of cutting-edge lightweight transfer learning models, and extensive testing on the animal-90 dataset, this research attempts to find models that achieve the best possible trade-off between classification accuracy and computational efficiency. The findings of this study are expected to provide valuable insights into the performance and efficiency of these models, offering a foundation for future practical implementations in wildlife conservation, ecological research, and related fields. Through this investigation, the thesis seeks to address the growing demand for robust and resource-efficient solutions in the realm of animal image classification, contributing to the broader goal of advancing technology for environmental monitoring and species identification.

1.1 Problem Statement

The growing necessity for accurate and efficient animal image classification in wildlife monitoring and ecological research is met with a significant challenge – the inherent computational constraints of resource-limited environments [12]. Traditional deep learning models, while powerful, often demand significant computational resources, making their deployment in scenarios such as wildlife cameras or mobile devices impractical. This disconnect underscores the pressing need for lightweight transfer learning models that can deliver robust classification performance with reduced computational overhead [17]. The current gap in the literature lies in the identification, empirical evaluation, and comparison among such models specifically tailored for animal image classification. Addressing this gap is critical to advancing the application of deep learning in wildlife conservation and environmental monitoring, where accurate species identification plays a vital part. Thus, the problem this thesis seeks to address is the development, evaluation, and comparison of lightweight transfer learning models to optimize the trade-off between computational efficiency and classification accuracy in the context of animal image classification.

1.2 Aims and Objectives

Aim:

This thesis aims to assess and compare the effectiveness of lightweight transfer learning models for animal image classification, with the overarching goal of identifying models that offer a balance between computational efficiency and classification accuracy.

Objectives:

- Conduct a thorough literature review to identify the best-performing lightweight transfer learning models for image classification.

- Select and experiment with the identified models on a diverse species dataset of animal images, considering factors such as model architecture, training techniques, and computational efficiency.

- Compare and analyze the performance of the selected models based on standard evaluation metrics and provide model selection guidelines based on the model performance.

1.3 Research Questions

RQ1: Which lightweight transfer learning architectures are most effective for classifying animals within the domain of image classification?

Motivation: The goal of this research question is to identify the most efficient lightweight transfer learning architectures in the domain of image classification. Traditional methods often fall short due to their time-consuming nature and limited accuracy, relying heavily on manual feature extraction and human input. Leveraging advanced transfer learning algorithms promises to overcome these barriers, offering more robust and precise classification capabilities. This study will conduct a thorough review of contemporary literature to pinpoint which architectures excel at efficiently and accurately processing images, thus advancing the field by enhancing both technological efficacy and practical application.

RQ2: How accurate are models trained using lightweight transfer learning architectures in animal image classification, and what practical implications do they hold?

Motivation: This research question helps in addressing the practical need for efficient and accurate image classification in the domains of wildlife conservation and ecological research. By evaluating the performance of selected algorithms on a diverse animal species dataset, this research seeks to provide insights into the effectiveness of lightweight models, which can be crucial in resource-constrained field environments. The results of this study can inform the selection of appropriate models for practical deployment, potentially enhancing the efficiency and success of real-world conservation and ecological research efforts.

1.4 Ethical, societal and sustainability aspects

1.4.1 Ethical Aspects

This thesis involves important ethical responsibilities as animal image data and models are utilized. All data was obtained ethically from public sources only, with a strict focus on subjects' welfare and privacy. Methodologies and results will be reported transparently to allow reproducibility and validation. Potential future applications in wildlife monitoring also require careful consideration of impacts on local ecosystems, biodiversity efforts, and communities near research areas. While striving for technological advancement, maintaining principled and accountable research conducted with a focus on environmental stewardship and social justice is paramount, to help ensure any technology ultimately benefits both wildlife and humanity.

1.4.2 Societal Impact

This thesis aims to further the societal good of wildlife conservation and environmental protection. By improving techniques for accurate species identification, the goal is to enhance ecological monitoring and support biodiversity preservation efforts. However, the possibility of biased or flawed models requires attention and oversight to ensure fair and just applications.

1.4.3 Sustainability Aspects

Developing computationally efficient models that perform well using minimal resources supports sustainability goals. Lighter models can reduce energy consumption, a key concern given the potential field-based uses of this technology. The focus on accuracy and efficiency also promotes responsive, adaptive tools for environmental management and protection. Overall, this research advocates technology that enhances sustainability through improved ecological understanding and stewardship of natural systems.

1.5 Outline

The outline of the thesis study is discussed below:

- **Introduction:** This chapter provides the introduction, outlining the background, problem statement, aims and objectives, research questions, and ethical considerations of the study.
- **Background:** This chapter presents the relevant background literature, covering key concepts in machine learning, transfer learning, model architectures, and performance evaluation metrics.
- **Related Work:** This chapter reviews prior related work to provide context around existing research and identify gaps to be addressed.
- **Methodology:** This chapter describes the methodology, including the literature review process, experimental setup, dataset, implementation details, and validation strategies.
- **Results and Analysis:** This chapter reports the results of both the literature review and experiments and key findings are analyzed. Also, provides model selection guidelines.
- **Discussion:** This chapter reflects on the results of each research question and considers implications for practical applications based on the results. Threats to validity were also discussed in this chapter.
- **Conclusions and Future Work:** This chapter concludes the study by summarizing the main contributions and proposing avenues for further research.

By outlining the flow of topics from introducing the research to analyzing results and drawing conclusions, this structure aims to convey the progression of the thesis.

Chapter 2

Background

2.1 Machine Learning

Machine learning is a data analytics technology that uses experience to learn. Machine learning technology creates decisions depending on what we tell the computers. Our reliance on databases for information storage and processing has grown as technology has advanced. The quantity and size of these databases are continually increasing. As a result of this expansion, manually extracting meaningful information becomes more challenging. As a result, we require semiautomatic and automated techniques to use, collect, analyze, and extract such data. Machine learning, data mining, and artificial intelligence methods and approaches have been demonstrated to be beneficial for these goals.

Machine learning algorithms can learn in two ways: (i) supervised learning (regression models, support vector machines, artificial neural networks, etc.).(ii) Unsupervised learning (clustering methods,) [18].

2.1.1 Supervised Learning

Supervised learning is a key area of machine learning in which algorithms are trained on labelled datasets. Each data point in the training dataset is paired with a known output or target in this technique. The basic goal of supervised learning is to allow algorithms to understand the link between input data and output data, which can then be utilized to make correct predictions or classifications when provided with fresh, previously unknown data. Supervised learning has two primary subcategories. These methods are frequently categorized based on their intended use [19]. Based on this feature, we may distinguish between two excellent families of supervised machine-learning techniques:

- 1. Classification
- 2. Regression

Classification:

A fundamental part of supervised learning is Classification. Its primary objective is to classify input data into specified classes or categories, especially when the output variable has a discrete class label. Classification methods such as logistic regression, decision trees, support vector machines, and neural networks are fully investigated. Practical applications and case studies of categorization models are provided, emphasizing their relevance in the real world. This is especially true when the output variable consists of discrete class labels. It handles scenarios such as distinguishing between spam emails, detecting fraudulent transactions, and categorizing photos into numerous object types.

Regression:

If the output parameter is continuous or real-valued, regression takes the focus. The study elaborates on regression approaches, with a focus on linear regression, polynomial regression, ridge regression, and support vector regression. To emphasize the practical usefulness of regression models, real-world applications are shown. Regression models' real-world applications are studied, revealing their use in areas such as banking, housing, and time estimate.

2.1.2 Unsupervised Learning

Unsupervised learning is the process of training algorithms on datasets without explicit supervision, implying that the data does not include labelled outputs [20]. Unsupervised learning requires algorithms to discover patterns, structures, or correlations in data on their own.

Clustering is a critical unsupervised learning issue in machine learning. Its major goal is to minimize data dimensionality and complexity by categorizing or grouping related data elements. This approach facilitates the finding of patterns, structure, and relationships within data, making it a vital tool in a wide range of fields and data analysis activities. Clustering techniques are used to investigate and comprehend data, allowing for more efficient analysis, visualization, and decision-making.

2.1.3 Semi-Supervised Learning

Semi-supervised learning has a distinct position within the machine learning spectrum, spanning the gap between supervised and unsupervised techniques. To train a model, this approach uses a small collection of labelled data and a large number of unlabeled data. The basic goal of semi-supervised learning is the same as that of supervised learning: to create a function that can reliably predict the output variable based on the input variables. However, what distinguishes semi-supervised learning is its use of a dataset that contains both labelled and unlabeled data [21].

Semi-supervised learning is especially useful when there is a large pool of unlabeled data but labelling it all is either prohibitively expensive or logistically difficult. Semi-supervised learning tries to use the benefits of both labelled and unlabeled data by combining this unlabeled data alongside a smaller collection of labelled data, achieving a balance between data efficiency and model accuracy. This method provides a realistic and cost-effective solution for many real-world machine-learning problems.

2.1.4 Reinforcment Learning

Reinforcement learning is a subfield of machine learning concerned with sequential decision-making in interactive contexts. An agent learns to execute behaviors to maximize cumulative rewards over time, which is the highlight of reinforcement learning. It is defined as a framework for learning optimal behavior through interaction with one's surroundings [22].

1. **Agent:** the decision-maker who engages with surroundings.

2. **Environment:** The external framework or structure within which the agent functions.

3. **States:** Representations of the current circumstance or environment setting.

4. **Actions:** In each stage, the collection of feasible movements or decisions that the agent can make.

5. **Rewards:**Numeric signals that offer feedback to the agent in response to its activities.

2.2 Deep Learning

A fundamental idea in artificial intelligence, neural networks are computer models inspired by the structure and function of the human brain. They are made up of linked layers of artificial neurons, also known as perceptrons, that are meant to process and learn from input. Each neuron takes input, computes a weighted sum of these inputs, and then applies an activation function to generate an output. Pattern recognition, categorization, and prediction are all activities that neural networks excel at. Their ability to learn and adapt from data, known as "training," enables them to solve complicated issues ranging from picture and audio recognition to natural language understanding.

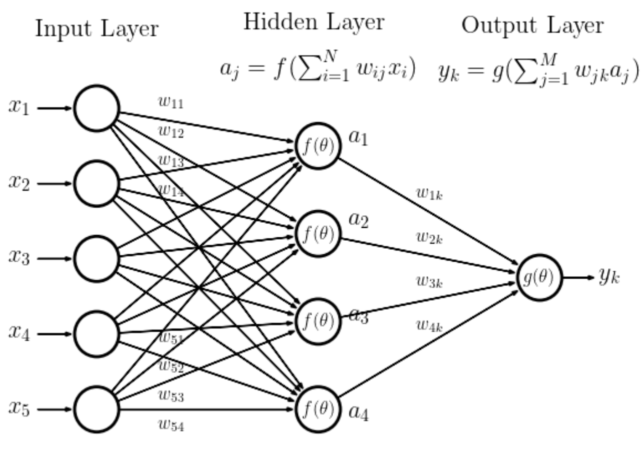

Figure 2.1: Deep learning architecture [1].

Deep learning, an advanced neural network architecture comprised of numerous hidden layers, is used in the creation and training of neural networks. They are composed of numerous layers (thus the word "deep"), each containing a huge number of artificial neurons. These networks excel in learning complex patterns and representations from big datasets. Deep neural networks (DNN) have had tremendous success in a variety of sectors.

2.2.1 Convolutional Neural Network

Convolutional Neural Networks (CNNs) are deep learning neural networks that are specially intended for processing and analyzing visual input such as photos and videos. It has transformed computer vision tasks by learning hierarchical features automatically from raw pixel data.

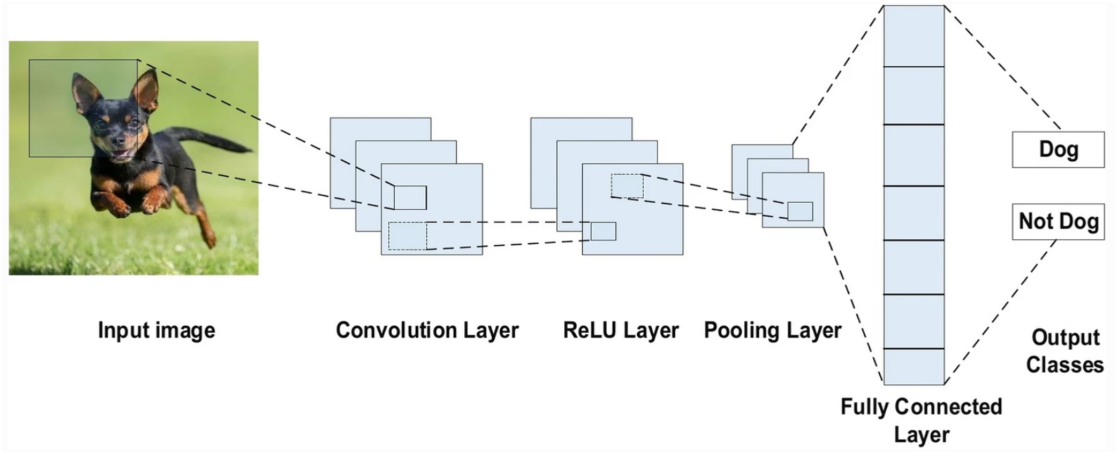

Figure 2.2: Convolutional Neural Network (CNN) architecture [2].

1. Convolution layers

CNNs extract local patterns and characteristics from input data using convolutional layers. Convolution is the process of detecting numerous characteristics in a picture by sliding tiny filters (also known as kernels) across it. In deeper levels of the network, these learned properties grow more complex.

2. Pooling layers

Pooling layers are used to reduce the spatial dimensions of the feature maps generated by convolutional layers. Pooling minimizes computational burden and makes the network more resilient to changes in input.

3. Fully connected layers

CNNs are similar to classic neural networks in that they typically comprise one or more fully linked layers following feature extraction. Pattern recognition and advanced decision-making are supported by these layers. To create predictions or classifications, they merge the acquired characteristics.

2.3 Transfer learning

Transfer learning in machine learning is like taking knowledge from one task and applying it to solve a related but different problem. It begins with using a pre-trained model on a sizable dataset, using the insights and features gained from that initial task to improve learning for a new task. By fine-tuning the pre-trained model on fresh data, it adapts its learned features to fit the specifics of the new information. This method reduces the need for extensive data and computational resources compared to starting from scratch, as shown in Figure 2.3. It becomes especially valuable when dealing with limited data or when computational capacity is restricted.
The procedure usually consists of two major steps:

1. Pre-training: A large dataset is used to train a model for a specific task, such as the recognition of images or natural language processing. In computer vision, for example, a model may be pre-trained on a large dataset such as ImageNet to recognize various objects in images.

2. Fine-tuning or Transfer Learning: Following pre-training, the model's knowledge is applied by reusing parts of the pre-trained model and adapting them to a new, often smaller, dataset or a related task. Using the new data, the model's parameters are adjusted or fine-tuned to improve performance on the target task

In the context of the Background chapter, the motivation for opting for lightweight transfer learning approaches over traditional techniques becomes essential. This choice is driven by the need to address computational constraints and make efficient use of available resources, particularly in scenarios such as ecological research and wildlife monitoring where robust models are required in resource-constrained settings. The emphasis on lightweight models aligns with the practical demands of real-world applications, promoting the effectiveness of transfer learning in scenarios with limited computational capacity.

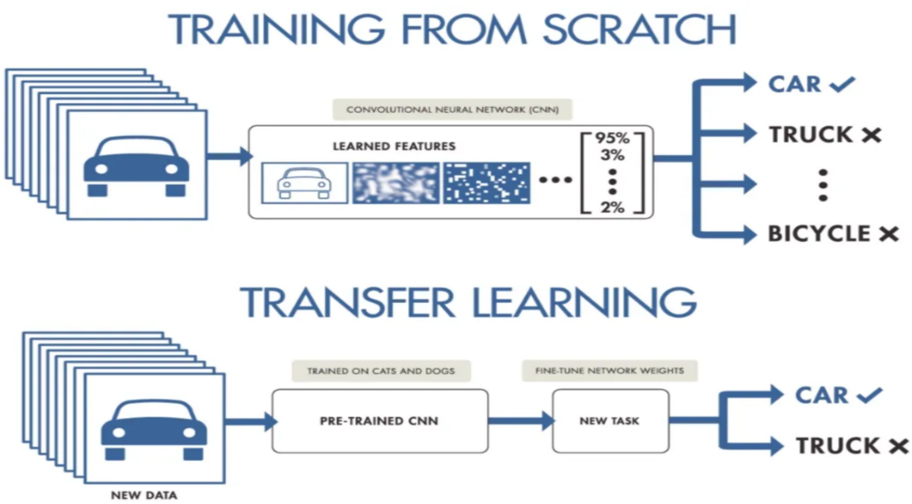

Figure 2.3: Transfer learning architecture [3].

2.3.1 Transfer Learning with Lightweight Models

In our study, we utilize different algorithms tailored to meet the distinct requirements and goals of each approach under investigation. These algorithm selections have been thoughtfully determined after thoroughly reviewing existing literature, as detailed in Section 4.1. It is essential to note that our algorithmic choices are founded on a comprehensive understanding of the strengths and limitations of each method, ensuring their suitability for our specific research objectives.

Moreover, our consideration of these algorithms encompasses their performance in handling various datasets and scenarios, aiming to maximize the efficacy and applicability of our research outcomes. Below, we explain the algorithms employed in our ongoing research.

MobileNetV2

MobileNetV2, developed by Google Inc., is a neural network architecture optimized for mobile and embedded devices. Its innovative design features inverted residual blocks with linear bottlenecks and utilizes depthwise separable convolutions to significantly reduce computational cost and model size [14]. This approach efficiently manages computational resources by expanding the input features, applying lightweight depthwise convolutions, and then compressing them, which conserves memory and accelerates inference. MobileNetV2 not only reduces the number of parameters and operations but also maintains competitive performance on tasks such as image classification, object detection, and segmentation. This makes it exceptionally suitable for applications in environments with limited computational and memory resources, demonstrating its versatility across a variety of vision-based tasks.

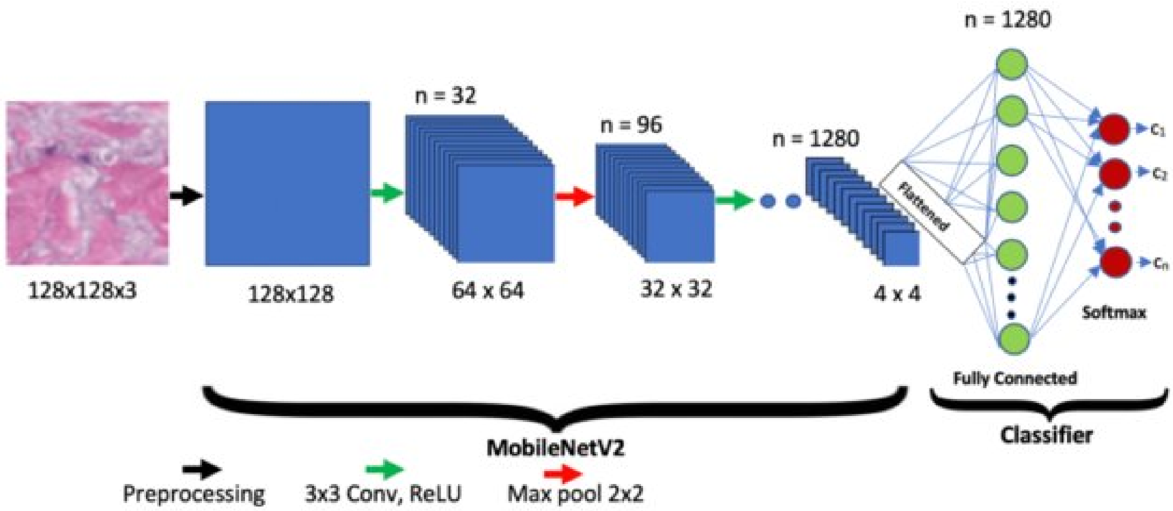

Figure 2.4: MobileNetV2 architecture [4].

EfficientNet B3

EfficientNet B3 is a pivotal convolutional neural network architecture, strategically positioned within the EfficientNet series. Striking a balance between computational efficiency and accuracy, B3 represents a mid-sized variant within this family. Its design integrates compound scaling, leveraging depth, width, and resolution adjustments to optimize performance without overburdening computational resources. Equipped with efficient building blocks like mobile inverted bottleneck convolution and Squeeze-and-Excitation (SE) blocks, it efficiently handles tasks such as image classification, object detection, and segmentation. Renowned for its adaptability through transfer learning, EfficientNet B3's pre-trained models serve as strong foundations for diverse computer vision applications. Its optimized structure, inclusive of swish activation functions and batch normalization, emphasizes feature recalibration and model efficiency. With competitive performance on ImageNet and an aptitude for resource-constrained environments, EfficientNet B3 stands as a compelling choice for various real-world deployments.

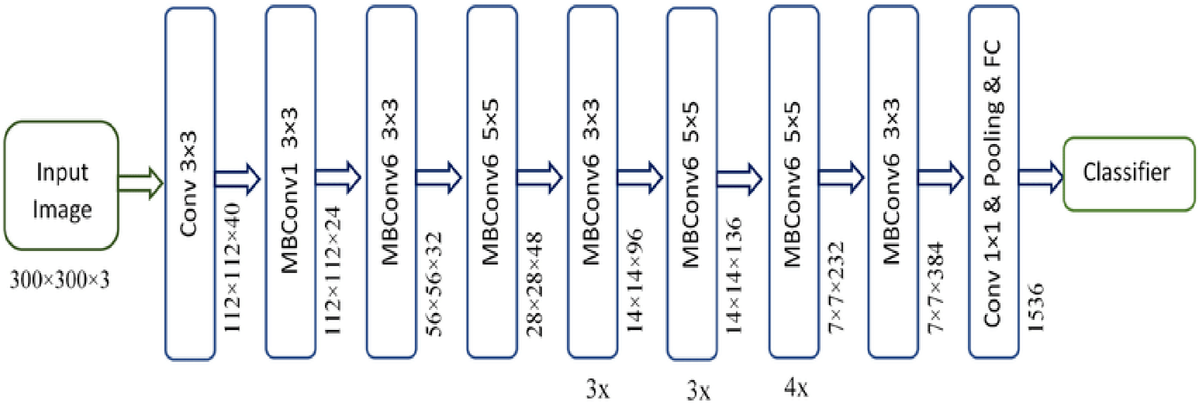

Figure 2.5: EfficientNet B3 architecture [5].

ShuffleNet

ShuffleNet, pioneered by Xiangyu Zhang, Xinyu Zhou, Mengxiao Lin, and Jian Sun, is tailored as an efficient deep neural network for mobile devices with limited computational resources, as elucidated in their work [6]. The architecture cleverly balances high accuracy with low computational load, ideal for real-time applications on memory and power-constrained devices. Central to ShuffleNet are pointwise group convolutions that segment convolutional features into groups to slash complexity, paired with a novel 'channel shuffle' operation that facilitates information flow between these groups, circumventing the potential bottlenecks of group convolutions. Additionally, it employs depthwise separable convolutions, akin to MobileNet, to further minimize computational demands. ShuffleNet also strategically downsamples early in the network, reducing spatial resolution promptly, thereby conserving computational effort in deeper layers, and enhancing overall efficiency suitable for the constrained environments of smartphones and similar devices.

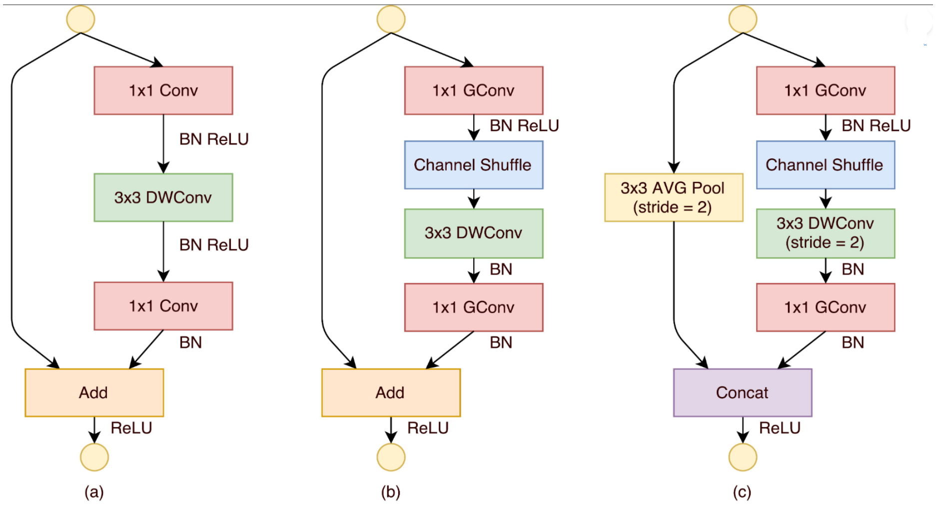

Figure 2.6: ShuffleNet Units. a) bottleneck unit with depthwise convolution (DW-Conv); b) ShuffleNet unit with pointwise group convolution (GConv) and channel shuffle; c) ShuffleNet unit with stride = 2 [6].

SqueezeNet

SqueezeNet, conceived by Iandola, Han, Moskewicz, Ashraf, Dally, and Keutzer, epitomizes a compact neural network designed to match AlexNet's accuracy with a model size 50 times smaller, as described in their seminal work [7]. The architecture is tailored for deployment on devices with constrained computational capacity, such as mobile and embedded devices, leveraging innovative design choices to compress its size without significant accuracy loss. Central to its architecture are the Fire modules, which comprise a squeeze layer of 1x1 filters that feed into an expanded layer combining 1x1 and 3x3 filters, efficiently condensing the network as shown in figure 2.7. This design is complemented by a reduced count of initial convolutional filters and a strategic delay in downsampling, promoting larger activation maps for heightened accuracy. Additionally, optional techniques like deep compression and quantization are employed to further compact the model, allowing SqueezeNet to achieve a fine balance between size, speed, and accuracy.

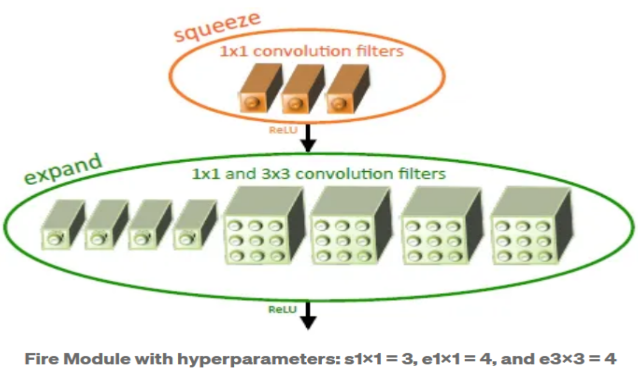

Figure 2.7: SqueezeNet architecture's Fire Module [7].

MnasNet

MnasNet, developed by Google researchers, is a family of mobile neural network architectures designed to optimize accuracy and efficiency for mobile platforms using Neural Architecture Search (NAS) [8]. This search strategy focuses on minimizing latency while maximizing accuracy, using a reinforcement learning algorithm that evaluates performance based on a combined metric of accuracy and computational efficiency tailored for mobile devices. The architecture leverages a Factorized Hierarchical Search approach as shown in Figure 2.8, organizing the search space into hierarchical blocks to streamline the search process and prioritize low latency and high accuracy. It employs MBConv, a mobile inverted bottleneck convolution similar to that used in MobileNetV2, featuring layers of depthwise separable convolutions that significantly reduce computational costs while maintaining effectiveness. Additionally, MnasNet incorporates Squeeze and Excitation (SE) blocks to adaptively recalibrate channel-wise feature responses, thereby boosting the network's representational capacity. The architecture also offers various scaling factors, allowing customization of the balance between speed and accuracy to meet different hardware or application requirements.

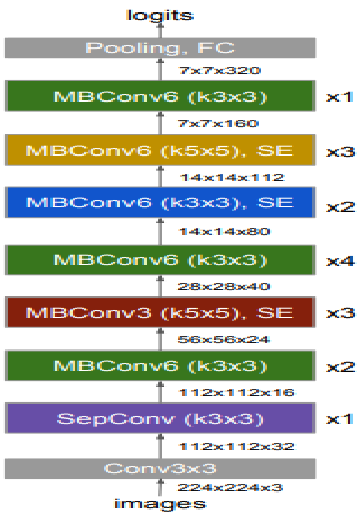

Figure 2.8: MnasNet architecture [8].

2.4 Model Validation Techniques

The need for a validation technique in machine learning arises from the essential goal of creating models that generalize well to new, unseen data. Validation allows us to assess how well our model performs on data it hasn't seen during training. Without validation, there is a risk of overfitting, where a model becomes too tailored to the training data and performs poorly on new examples. By using validation techniques like k-fold cross-validation, we obtain a more reliable estimate of a model's performance across different subsets of the data, helping ensure its ability to make accurate predictions on diverse inputs. This process enhances the robustness and generalization capability of machine learning models, contributing to their effectiveness in real-world scenarios.

K-Fold Cross Validation: K-fold cross-validation is a robust technique used to assess the performance of a machine learning model by partitioning the dataset into 'k' subsets or folds. The process involves iteratively training and evaluating the model 'k' times, with each iteration using a different fold as the validation set while the remaining folds serve as the training set. This approach provides a more comprehensive evaluation, as each data point is part of the validation set exactly once [23].

In each iteration, the model learns from different parts of the data, helping to ensure that its performance is representative across the entire dataset. The final evaluation metric is then averaged over the 'k' iterations, yielding a more reliable estimate of the model's generalization ability.

K-fold cross-validation is particularly useful when the dataset is limited, as it maximizes the use of available data for both training and validation. This technique helps mitigate the risk of overfitting and provides a more accurate assessment of a model's performance, contributing to the credibility of its results in diverse scenarios.

2.5 Performance Metrics

To understand how well the algorithms worked, different ways of measuring their performance were used [24]. This study looked at various measures to compare how well the algorithms performed. They checked things like accuracy, recall (which is about finding all the right things), precision (which is about being accurate when saying something is right), and F1 score (which considers both precision and recall together)

1. Accuracy

Accuracy stands as a fundamental metric in assessing the correctness of a model by gauging its accurate predictions concerning all examples. Despite offering an initial overview of performance, accuracy can become misleading, especially when dealing with imbalanced datasets. The mathematical representation for accuracy is defined as follows [24]:

$$Accuracy = (Tp + Tn)/\ (Tp + Tn + Fp + Fn) * 100$$

where,

- Tn = True negative
- Tp = True positive
- Fp = False positive
- Fn = False negative

2. Recall

Recall, also known as sensitivity, measures a model's ability to locate all pertinent instances. It particularly emphasizes true positives, illustrating how effectively the model captures actual positives. This metric holds significance in scenarios where the cost of missing positive results is considerably high [24].

The mathematical representation for recall is given by:

$$Recall = Tp\ /\ (Tp + Fn)$$

3. Precision:

Precision evaluates the accuracy of positive predictions by comparing true positives to the instances predicted as positives. It assumes crucial importance in cases where minimizing false positives is imperative [24].

The formula for precision is articulated as:

$$Precision = TP\ /\ (TP + FP)$$

4. F1 Score

The F1 score, categorized as a machine learning metric within classification tasks, integrates precision and recall metrics into a unified measurement. It consolidates both precision and recall into a singular metric, offering a comprehensive evaluation of a model's performance [24].

The mathematical representation for the F1 Score is given by

$$F1\ score = 2 * (precision*recall)/(precision+recall)$$

Chapter 3

Related Work

Transfer Learning in Animal Image Classification

Several notable studies have propelled animal image classification through diverse methodologies. Hu et al. [25] achieved impressive accuracies of 97% for individual animals and 92% overall by employing transfer learning with ResNet18, yet their exclusive focus on this network calls for further exploration of alternative deep learning models. Similarly, DFC Huertas et al. [26] showcased robust classification accuracy with a ResNet101-based model, outperforming established architectures like GoogleNet-2014, Vgg16, and ResNet50. Their emphasis on the significance of residual units for enhanced knowledge transfer between related tasks and the open sharing of TensorFlow 2 code promote continued exploration of varied architectures and loss functions in smart farming and medical imaging. Moreover, Sanghvi et al.'s [27] Fauna Image Classification system utilizing Convolutional Neural Networks (CNNs) yielded an impressive 91.84% accuracy, significantly aiding ecologists in habitat studies. Leveraging VGG16 and TensorFlow, their research highlighted the efficacy of the ReLU activation function and VGG16 model in fauna image classification, showcasing CNNs' potential in advancing ecological studies. These studies collectively underscore the importance of diverse model exploration and the pivotal role of CNNs in precise animal image recognition, supporting environmental research and habitat conservation efforts.

Conversely, Shahram Taheri et al. [28] introduced an innovative animal biometric system centred on image and video data analysis using machine vision techniques. Their method fused convolutional neural network (CNN) features with kernel Fisher analysis (KFA-"It performs non-linear feature extraction and dimensionality reduction using kernel methods for enhanced class separability") based descriptor features, demonstrating superior classification accuracy (95.31%) on animal faces compared to traditional techniques. Leveraging the strengths of CNN's feature extraction capability and KFA's unique approach, their system, based on the VGG-16 model fine-tuned on the LHI-Animal-Faces dataset, surpassed existing state-of-the-art systems, showcasing the efficacy of score-level fusion in enhancing animal face classification. Additionally, Y H Sharath Kumar et al. [29] introduced a novel animal classification method employing graph cut-based segmentation and classifier evaluation on a proprietary dataset. Their findings highlighted the effectiveness of K-Nearest Neighbors (KNN) in achieving higher classification accuracy compared to Probabilistic Neural Network (PNN) across varying block configurations. This emphasized the superiority of KNN in animal classification within diverse settings, underscoring its performance over alternative classifiers.

In the integration of innovative methodologies in aquatic animal image classification, Yuan, H et al. [30]employed a data augmentation via DCGAN and leveraging transfer learning techniques, leading to an impressive 97.9% accuracy. Experimentation with varied network structures and trainable parameter proportions showcased the effectiveness of pre-trained models like Inception and ResNet. Inception demonstrated stability and high parameter utilization, while ResNet offered expedited training despite marginally lower expressive capacity. Conversely, in predator animal detection, F Alharbi et al. [31] introduced a distinct approach centred on crucial features such as eyes and ears, utilizing SVM and MLP for classification. Their method exhibited promising accuracies, emphasizing its potential for identifying hazardous animals in outdoor settings, enhancing safety, and aiding livestock protection. This research underscores the social significance of animal species classification while prompting future exploration into expanding datasets and incorporating deep learning techniques for heightened accuracy in diverse real-world scenarios.

Wildlife Image Classification

Hung Nguyen et al., [32] introduced an innovative approach to automate wildlife monitoring by applying deep learning techniques to camera trap images. Their system achieved a remarkable 96.6% accuracy in detecting animal-containing images and approximately 90.4% accuracy in identifying common species in South-central Victoria, Australia, exhibiting robustness in varied experimental settings. Future directions involve refining the system through dataset expansion, deeper CNN models, and transfer learning to address imbalanced data. This aims to enhance citizen science-based wildlife monitoring initiatives like the Wildlife Spotter project, offering efficient solutions for ecological research, conservation, and management decisions. Similarly, Willi et al., [33] combined deep learning with citizen scientists' efforts for wildlife image classification, attaining high accuracies for image differentiation and species recognition. Transfer learning notably improved accuracy, showcasing the potential for efficient data processing in large-scale camera trap studies and aiding ecology researchers while reducing human effort significantly.

In the realm of wildlife image classification, Zhang et al., [34] present an innovative method addressing limited training data challenges for rare species classification. Through manual and CycleGAN-based augmentation, they expand datasets, enabling a ResNet50 model to achieve 92.2% accuracy with only 10 images per class. Comparative experiments highlight CycleGAN's superiority over other augmentation methods, showcasing its potential for enhancing limited training data. Conversely, Zualkernan et al., [35] explore IoT and deep learning integration for real-time animal image classification. Utilizing various models trained on 66,400 images, their study identifies Xception as a standout performer with 96.1% accuracy. Deployment on edge devices reveals trade-offs between performance and power consumption, emphasizing the significance of model selection and device optimization. These studies collectively advance wildlife image classification and real-time monitoring, emphasizing the synergy between deep learning techniques and IoT platforms.

Lightweight Architectures for Image Classification

Bin Jiang et al., [36,37] introduced two innovative approaches for animal image classification, firstly they used the Bilateral Convolutional Network (BCNet-"It is a deep learning architecture that performs bilateral learning via extracting and fusing edge-based representations to improve image classification performance") in the paper [36] that focused on lightweight animal recognition, balancing accuracy and model size for mobile devices. It incorporated two sub-networks emphasizing object location and feature extraction, achieving an 85.6% top-1 accuracy on the Animals with Attributes (AWA) dataset. Their exploration of network depth revealed that increased depth didn't significantly boost accuracy but led to larger models. BCNet displayed efficiency, being 40.84% smaller than ResNet101 and 29.85% smaller than ResNet152, suitable for mobile applications. In contrast, they proposed in the paper [37], the Multiprocess Convolutional Network (MPNet-"It is a multi-scale convolutional neural network architecture that processes inputs through multiple parallel convolutional processes to efficiently capture representations at different scales") addressing computational demands in animal image classification. This approach, leveraging weight-sharing strategies and dual sub-networks, attained an impressive 87.54% top-1 accuracy on the AWA dataset with a minimal 33.57MB parameter size. Outperforming existing methods in both accuracy and computational efficiency, MPNet underscores the significance of lightweight transfer learning in resource-constrained settings. They suggested potential extensions of this approach to object detection, fine-grained recognition, and pointwise localization, highlighting its versatility across various applications.

In marine aquaculture, Liu et al. [38] present a method integrating an embedded system with deep learning, using MobileNetV2 and transfer learning for efficient real-time marine animal image classification. Their approach, employing an underwater robot and an embedded device, constructs a MobileNetV2-based convolutional neural network (CNN) model for processing marine animal images, achieving 92.89% validation accuracy compared to InceptionV3 and MobileNetV1. MobileNetV2's 40M size proves optimal for embedded devices, showcasing lightweight transfer learning's potential in resource-constrained environments. Similarly, Kondaveeti et al. [39] utilize MobileNetV2 for bird species recognition, attaining 96.79% accuracy on South Indian bird species and 90.91% on a larger dataset with 200 species, highlighting its efficiency in real-time bird identification and population monitoring. Meanwhile, Shahi et al. [40] focus on fruit image classification, combining MobileNetV2 with an attention module to capture object-based and semantic information, achieving 95.75% to 96.74% accuracy across various datasets. This approach balances fewer trainable parameters with enhanced accuracy, demonstrating potential in automated classification for fruit-related industries. The studies collectively highlight the effectiveness of lightweight transfer learning, suggesting future exploration of alternative lightweight architectures, advanced data augmentation methods like Generative Adversarial Networks (GANs), and efficient operation in mobile or Internet of Things (IoT) setups.

In the realm of image classification, S Karthigai Selvi [41] introduces AniDetNet, utilizing EfficientNet-B3 with transfer learning to classify various animals. Achieving a notable 98.5% accuracy, this system emphasizes risk avoidance in classification using Convolutional Neural Networks (CNNs). It successfully recognizes and segments 150 animals, crucial for tourism, forestry departments, and knowledge seekers, showcasing modifications in training mechanisms that substantially improve recognition rates. Meanwhile, Haomin Liu [42] addresses small sample image classification challenges through transfer learning, employing data augmentation and MobileNet-V2. Their approach achieves a remarkable 95.16% accuracy on the Animals-10 dataset, emphasizing transfer learning's efficiency in leveraging pre-trained features and accelerating network convergence. Liu's study, based on a Kaggle competition dataset, aims to further enhance recognition rates and practical utility through target detection functions and diverse dataset experiments, contributing significantly to lightweight transfer learning methodologies for efficient image classification with limited samples.

Research gap

The current body of research provides an extensive overview of methodologies used in animal recognition and wildlife monitoring, including techniques such as convolutional neural networks (CNNs), transfer learning, and lightweight models like the Bilateral Convolutional Network (BCNet). However, a notable gap exists in the comparative analysis of lightweight transfer learning models, particularly for animal image classification with datasets encompassing a broad spectrum of species classifications.

To address this gap, a comprehensive investigation into existing research on lightweight models is crucial. Identifying, assessing, and comparing the performance of these models is vital. The main objective is to identify the best-performing models in the lightweight architecture category from existing literature, evaluate them on a selected dataset, and compare their performance in terms of accuracy and efficiency to determine their practical suitability in real-world scenarios with constrained resources. This research aims to bridge the gap in our understanding of lightweight transfer learning models through comparative performance analysis and provide practical guidance on selecting effective models for animal image classification tasks in resource-constrained environments.

Chapter 4

Methodology

In this chapter, we describe how we approached our research questions and connected them to our main goal. To answer these questions, we used specific methods. While we reviewed existing studies for RQ1, and for RQ2, we experimented. We'll explain our decisions and how they align with each question, providing details on our methods and reasoning behind them.

Justification for methods selected:

Literature Review for answering RQ1: To address RQ1, a comprehensive literature review is chosen as the primary method. This review involves a meticulous evaluation of various lightweight algorithms specialized in classifying images. By critically analyzing papers that discuss the performance and accuracy of different models in predicting and categorizing animal images, the aim is to identify the most efficient and robust lightweight architectures. This thorough investigation intends to illuminate the capabilities of these models, demonstrating their potential to outperform traditional methods. The ultimate goal is to justify the selection of optimal lightweight architectures for image classification based on a solid foundation of existing research.

Experimentation for answering RQ2: For RQ2, the chosen method involves rigorous experimentation with a diverse dataset encompassing various animal species. Through extensive training and testing of different pre-trained lightweight models on this representative dataset, the research aims to assess the performance of these models in animal image classification. The evaluation process focuses on identifying models that exhibit superior accuracy and efficiency, specifically for practical deployment in wildlife conservation and ecological research scenarios. The overarching objective is to bridge theoretical advancements with real-world conservation needs, providing valuable insights for practical applications and contributing to the effective integration of technology in ecological research and conservation efforts.

Research Approach:

The research design of this study encompasses a comprehensive exploration of existing literature followed by a structured experimentation process.

Literature Review:

The study begins with an extensive literature review, delving into various scholarly works and research articles related to lightweight deep learning architectures, image classification, and model optimization. The review synthesizes key findings, methodologies, and insights from the literature, providing a foundation for understanding the state of the art in the field.

Experimental Design:

Following the literature review, the research transitions into the experimentation phase. The primary objective is to assess the efficacy of selected lightweight architectures, in the context of animal image classification tasks. Below are the steps followed in the experimentation.

- Data Preprocessing: The first step involves meticulous data preprocessing, where the raw image dataset undergoes labeling, resizing, and augmentation. Labeling assigns specific class labels to each image, enabling the models to recognize and classify animals accurately. Resizing ensures uniformity in image dimensions, and augmentation techniques enhance model robustness.

- Model Training: The core of the experimentation lies in training the selected models. The study employs a transfer learning approach, leveraging pre-trained knowledge from the selected architectures. The models are compiled with appropriate optimizers, loss functions, and evaluation metrics. Training involves the use of a train-test split and data augmentation to enhance model generalization. Training also involves Fine-Tuning the model further to refine their understanding of the dataset and Hyperparameter Tuning to enhance the models' efficiency and effectiveness.

- Model Evaluation: The performance of the trained models is rigorously evaluated using k-fold cross-validation, specifically 5-fold, to ensure a robust assessment. Metrics such as accuracy, precision, recall, F1 score, loss, average inference time per batch, and the number of parameters are employed to evaluate the models' classification capabilities. To provide a comprehensive and fair comparison of the models, the Friedman test is conducted. This non-parametric test assesses the differences in performance metrics across multiple models, determining if there are statistically significant differences in their rankings.

  Furthermore, a combined metric approach is utilized to aggregate the performance metrics into a single score for each model. This approach involves normalizing individual metrics and calculating a weighted average, considering the relative importance of each metric. This combined metric offers a holistic view of each model's performance, facilitating a more nuanced comparison.

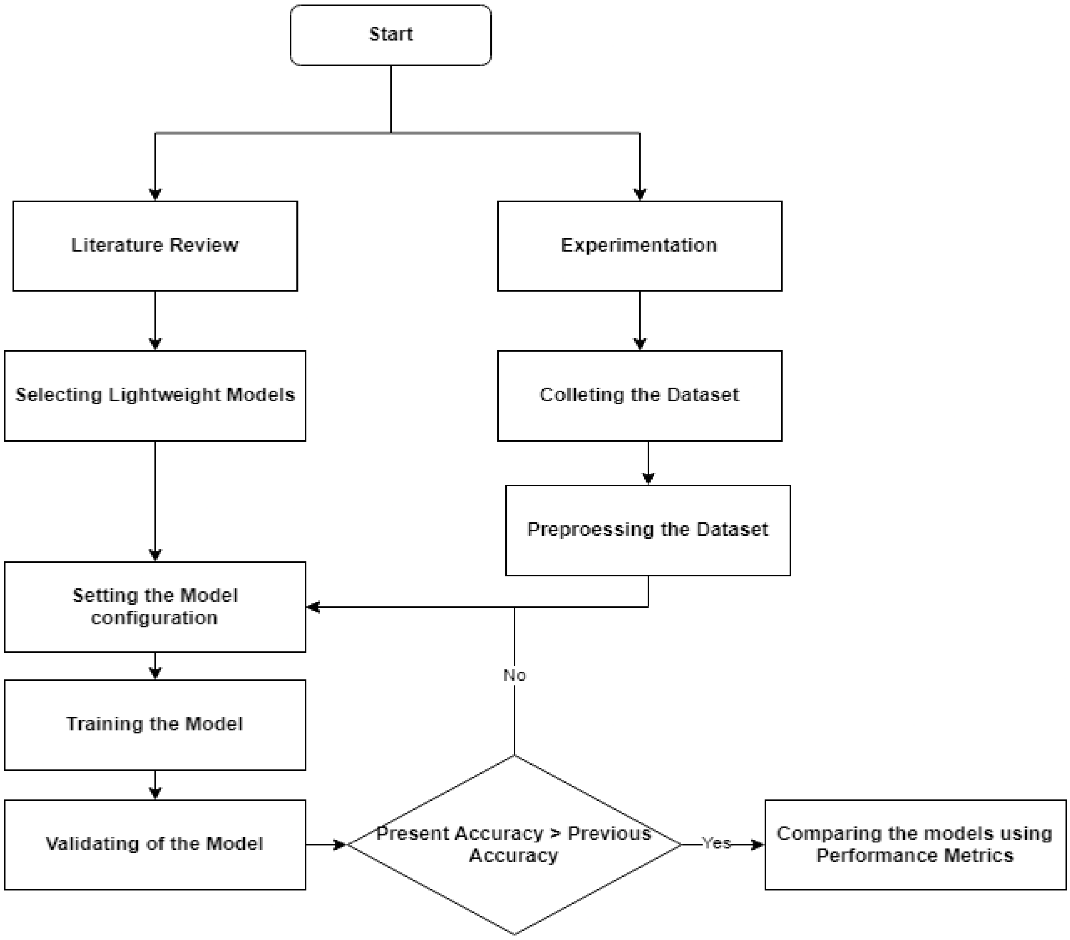

Figure 4.1: Research Approach Flowchart .

In summary, the research design combines a thorough literature review with a systematic experimental approach, aiming to contribute valuable insights into the application of lightweight deep learning architectures for image classification, particularly in the domain of animal images.

4.1 Literature Review

To better understand our research topic, conducting a thorough literature review is crucial. This involves gathering information from recent studies to establish a strong foundation for our investigation. Our first research question RQ1 aims to determine which lightweight transfer learning architectures are most effective for classifying animals within the domain of image classification. To address this question, we're utilizing a literature review as our method. By analyzing existing studies, this review helps us identify different algorithms used in this specific area. It's an essential step in tackling this issue. We're implementing a comprehensive search strategy to find relevant research articles from databases like Google Scholar, ScienceDirect, IEEE Xplore, and others, as depicted in Figure 4.2.

To gather information for our literature review, we followed a snowballing method.

This approach involved looking at related papers to expand our sources. We searched both forward(Citations) and backward(References) to find more relevant materials. Our exploration extended across diverse scientific databases such as IEEE Xplore, ScienceDirect, ResearchGate, and similar platforms. Papers were chosen based on their references, encompassing studies related to image classification using transfer learning techniques, and lightweight architectures. Our assessment criteria for relevance considered specific factors integral to the research.

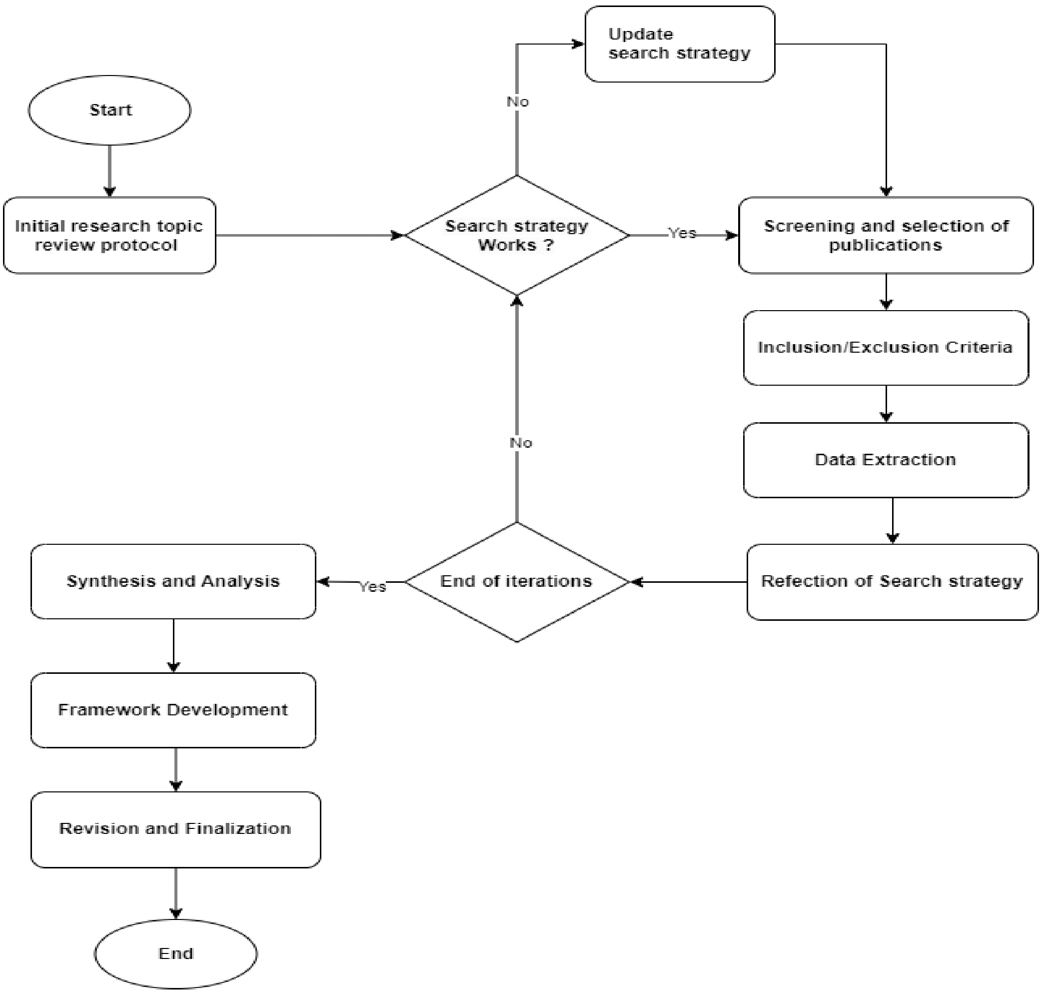

Figure 4.2: Literature review architecture.

4.1.1 Inclusion and Exclusion Criteria

- **Inclusion criteria:**
  - Identified closely related works distributed through conferences, journals, and books.
  - Focused on evaluating items published in and after 2019 to stay updated with recent technical advancements.
  - The associated works are available in BIBtext format for easy referencing.

- Articles published exclusively in English were included for consideration.

- **Exclusion criteria:**
  - Non-English publications were likewise omitted from the review procedure.
  - Studies that did not align with the preferred methodologies or analytical approaches required for the research objectives were excluded.
  - Articles that didn't provide enough information or data, making it hard to understand or use their findings, were excluded.
  - No duplicate publications to avoid redundancy during the literature review process.

To address research question RQ1, this study utilizes a literature review method. Referring to the figure above, our initial step involves selecting relevant papers focusing on our topic. Our goal is to explore and assess lightweight transfer learning architectures used for constructing models in animal image classification. This involves starting with a review of previous research papers in the domain of image classification to understand the various methods used. The findings shared in this study gather important insights from multiple research sources, laying a strong foundation for the later stages of our investigation.

4.1.2 Selected Literature

Paper Title	**Conclusions drawn**
Comparative study and analysis on skin cancer detection using machine learning and deep learning algorithms [43].	The study explores skin cancer prediction using machine learning (ML) and deep learning (DL) techniques on dermoscopic images. Various models were tested, with Random Forest achieving 87.32% accuracy and MobileNetv2 reaching 97.58% accuracy with augmentation. Customized CNN architectures, particularly those with 5 and 3 layers, achieved high accuracies of 97.72% and 98.02%, respectively, on augmented datasets. Transfer learning models, notably ensemble models and MobileNetv2 displayed promising performance, indicating potential clinical integration pending further validation.

Paper Title	Conclusions drawn
Comparative analysis of imaging diagnostic models for tubular basophilia and mineralization of kidney [44].	This research delved into diagnosing tubular basophilia and mineralization, common toxicological lesions in laboratory animals, through whole-slide imaging and deep-learning techniques. Employing transfer learning with Convolutional Neural Network (CNN) models like InceptionV3 and Xception, the study aimed to achieve a comprehensive understanding of toxicopathological features. Comparative analysis of MobileNetV2, Xception, and InceptionV3 highlighted MobileNetV2's superior accuracy (98.57%) in lesion classification.
Harnessing the Power of Transfer Learning in Sunflower Disease Detection: A Comparative Study [45].	This research explored deep learning models for sunflower disease classification, employing five popular architectures. Among these models, EfficientNetB3 achieved the highest accuracy of 0.979, outperforming others like AlexNet, VGG16, InceptionV3, and MobileNetV3, which achieved accuracies of 0.865, 0.965, 0.954, and 0.969, respectively. The study highlights the effectiveness of deep learning in accurately detecting and classifying sunflower diseases, offering valuable insights for timely agricultural management strategies. EfficientNetB3 and MobileNetV3 emerged as top-performing models due to their high accuracy and efficiency.
Ocular Disease Identification Using Deep Learning Techniques [46].	This study proposed EfficientNetB3 for ocular disease identification, employing Transfer Learning (TL) and Convolutional Neural Networks (CNNs) on fundus images. With a dataset of 4000 fundus images sourced from various repositories, they validated the model on 400 images, achieving an accuracy of 93.8%. EfficientNetB3 was fine-tuned to optimize performance, surpassing VGG19 and ResNet50 models, displaying an accuracy of 91.50%, precision of 92.69%, recall of 91.50%, and an F1-score of 91.56% in diagnosing cataract, diabetic retinopathy, and glaucoma. The study highlights EfficientNetB3's superiority for accurate and efficient ocular disease diagnosis.

Paper Title	Conclusions drawn
An Efficient Pneumonia Detection from the Chest X-Ray Images. [47].	The study delves into automated Pneumonia diagnosis amid the COVID-19 pandemic using deep learning techniques. It focuses on classifying pneumonia types from X-ray images, reducing the strain on human resources. VGG16, VGG19, ResNet50, MobileNetV1, and EfficientNetB3 were compared for image classification based on accuracy, f-measure, and AUC metrics. EfficientNetB3 outperformed others, achieving 88.78% accuracy for both binary and ternary classifications. These results represent the best-reported accuracies within the dataset, showcasing EfficientNetB3's potential for automating Pneumonia diagnosis in web or mobile applications.
Comparative Study of Convolutional Neural Network Feature Extractors Used for COVID-19 Detection from Chest X-Ray Images [48].	The research evaluates feature extractors for COVID-19 detection using PA chest X-ray images, focusing on ground-glass opacities. Among the studied extractors like MobileNetV2, ResNet50V2, ResNet152V2, Xception, VGG16, and DenseNet12, MobileNetV2 exhibits superior performance, achieving 91.28% accuracy and a 0.891 F-1 score. MobileNetV2's efficiency, with 2,430,403 total parameters and an inference time of 0.2737s, suggests its suitability as an optimal algorithm for COVID-19 classification in PA chest X-ray images.
Individual Beef Cattle Identification Using Muzzle Images and Deep Learning Techniques [49]	This research tested several architectures and assessed deep learning models for recognizing individual beef calves from muzzle photos. The performance varied notably. MobileNetV2 was solid with 91.3% accuracy, though not the best. EfficientNet B3 only achieved 60.0% accuracy, showing variable performance with different data sets. MnasNet, designed for mobile use, had 57.6% accuracy, indicating less suitability for muzzle image recognition. In contrast, SqueezeNet excelled with 95.9% accuracy due to its small size and speed, ideal for certain deployments. ShuffleNet, however, performed poorly in this application with only 1.3% accuracy. This study underscores the varying effectiveness of models in cattle identification, with some models like SqueezeNet outperforming more complex ones in specific scenarios

Paper Title	Conclusions drawn
A Method of Body Condition Scoring for Dairy Cows Based on Lightweight Convolution Neural Network [50]	The ShuffleNet model outperforms other models for cow body condition grading, with an accuracy rate of 98.2%. This model outperforms the original ShuffleNetV2 and other popular models such as MobileNetV2, EfficientNetV1, and ConvNeXt in terms of performance. Despite its improvements, it remains lightweight and portable, making it excellent for practical usage in dairy farming.
Study on transfer learning capabilities for pneumonia classification in chest-x-rays images [51]	This study has Several deep learning models that were evaluated with an emphasis on COVID-19 to determine how well they could diagnose pneumonia in chest x-ray pictures. In medical imaging tasks, MobileNetV2 demonstrated strong performance with an F1-score of 82.48%, a precision of 83.78%, and a sensitivity of 81.91%. By comparison, MnasNet's performance was less successful, showing difficulties with adaption in this situation with a precision of 69.50%, sensitivity of 56.49%, and F1-score of 54.60%. SqueezeNet could not perform up to par, achieving just 20.73% accuracy, 27.49% sensitivity, and an F1-score of 19.70%. This suggests that SqueezeNet may have architectural limits when it comes to complicated classifications like medical imaging. ShuffleNet demonstrated good competence in medical picture analysis, achieving commendable results with an accuracy of 83.19%, sensitivity of 78.04%, and an F1-score of 79.33%.
MnasNet: Platform-Aware Neural Architecture Search for Mobile [8]	This study examines how neural network designs may be optimized for mobile contexts, including specifics on each model's performance. For example, MobileNetV2 attains a top-1 accuracy of 72.0% with an inference latency of 75 ms; however, its architectural search does not incorporate direct latency measurements. By deliberately taking into account real-world latency during design, MnasNet outperforms this and produces superior results. At a latency of 78 ms, MnasNet-A1 attains a top-1 accuracy of 75.2%; MnasNet-A2 improves accuracy to 75.6%; and MnasNet-A3 surpasses this with a top-1 accuracy of 76.7% at a latency of 103 ms. This illustrates how MnasNet can effectively balance acceptable latency and great accuracy on mobile devices.

Paper Title	Conclusions drawn
Automatic detection of bacilli bacteria from Ziehl-neelsen sputum smear images [52]	In this work, SqueezeNet was used as a classifier and successfully identified bacilli bacteria in Ziehl-Neelsen stain photos with an impressive 97% overall accuracy. The model's compact size of 4.8 MB allowed it to interpret complicated picture data with great efficiency, offering a workable alternative for automated medical image analysis in resource-constrained settings.
Comparative Analysis of Lightweight Pre-Trained CNN Models for Coffee Bean Roasting Level Identification [53]	In this study, SqueezeNet, ShuffleNet, MobileNetV2, and MnasNet (NasNet Mobile) all identified coffee bean roast levels with 100% accuracy. Notable for its effectiveness and low number of parameters, SqueezeNet has a small model size of only 2.7 MB, which makes it perfect for embedded devices with little memory and processing capacity. ShuffleNet improves efficiency by channel shuffling, which maximizes feature map communication. Its slightly larger model size is 3.3 MB. Despite being bigger (8.7 MB), MobileNetV2 maintains good accuracy by using depthwise separable convolutions, making it appropriate for mobile devices. The biggest, MnasNet (16.1 MB), is meant to maximize accuracy and latency, producing a bigger yet incredibly efficient architecture.
Plant Disease Classification using Lite Pretrained Deep Convolutional Neural Network on Android Mobile Device [54]	In the research paper, the efficiency of MnasNet and MobileNet for categorizing plant diseases on mobile devices was assessed. With a significant accuracy of 92.83%, MobileNet stands out for its effectiveness and small model size, making it a good choice for real-time applications on devices with constrained processing power. With an accuracy of 94.87%, MnasNet fared somewhat better than MobileNet. This demonstrates MnasNet's outstanding ability to manage picture classification jobs efficiently, even with mobile technology limitations. High accuracy rates were shown by both models, highlighting their potential as dependable choices for mobile applications requiring strong picture recognition skills.

Paper Title	Conclusions drawn
Lightweight Transfer Learning Models for Ultrasound-Guided Classification of COVID-19 Patients [55]	The study examined the performance of lightweight models—MobileNet, ShuffleNet, and MnasNet—in identifying the lung health status of ultrasound pictures. Both ShuffleNet and MobileNet surpassed 97.00% classification accuracy. MobileNet is well-known for its simplified architecture and performance, which makes it perfect for real-time mobile applications. Similarly, ShuffleNet works well in resource-constrained contexts by reducing processing needs through the use of channel shuffling and pointwise group convolutions. With an exceptional accuracy of 99.00%, MnasNet outperformed these models, proving its capacity to balance accuracy and speed efficiently.
Classification of Fruit Plants Leaf and Comparative Analysis of Machine Learning and Deep Learning Algorithms [56]	The research presented the accuracy of several models, demonstrating the efficiency of lightweight CNN models in challenging picture classification tasks such as fruit plant leaf identification. With an exceptional accuracy of 99.96%, ShuffleNet performed exceptionally well. This was facilitated by its architectural design, which was designed for performance and efficiency. As a result, it was a great fit for jobs requiring in-depth picture analysis. However, SqueezeNet managed to get a commendable accuracy of 93.68%. Though it was somewhat surpassed by ShuffleNet, it showed a high competence in handling complicated picture classifications despite its compact architecture intended to decrease memory use.
Comparison of Deep Learning Models for the Classification of Noctilucent Cloud Images [57]	In this research, SqueezeNet, ShuffleNet, and MobileNetV2 were assessed, and each obtained an F1 score of 0.95. SqueezeNet's 21.81 MB tiny size was highlighted as a feature that maximizes efficiency for systems with constrained hardware. ShuffleNet was optimized for mobile applications by using pointwise group convolution and channel shuffling to ensure good accuracy with low computing needs, despite its small size of 35.13 MB. With a size of 24.90 MB, MobileNetV2 uses depthwise separable convolutions to efficiently balance accuracy and latency, meeting the demands of mobile devices for computational efficiency.

Table 4.1: Selected Literature

4.2 Experimentation

In response to the research question "How accurate are models trained using lightweight transfer learning architectures in animal image classification, and what practical implications do they hold" a systematic approach was adopted. The first step involved an extensive literature review to identify state-of-the-art lightweight transfer learning models suitable for image classification. The literature review guided the selection of models that demonstrated promise in achieving accuracy and computational efficiency.

Based on the findings and results from the literature review (as addressed in Research Question 1), MobileNetV2, EfficientNetB3, ShuffleNet, SqueezeNet, and MnasNet were recognized as standout choices for the experiment. These models were chosen for their proven effectiveness in image classification tasks and their lightweight architectures, making them particularly suitable for resource-constrained environments. The experiment, detailed in the following sections, involves training and fine-tuning these selected models on a diverse species dataset of animal images, assessing their accuracy, and uncovering the practical implications they may hold for real-world applications in wildlife conservation and environmental monitoring.

4.2.1 Friedman Test Analysis

This analysis aims to statistically compare the performance of the lightweight models MobileNetV2, EfficientNetB3, ShuffleNet, SqueezeNet, and MnasNet based on the accuracy metric using a Friedman test. The analysis is conducted on the accuracy results obtained from 5-fold cross-validation.

- Null Hypothesis (H_0): There is no significant difference in performance among the lightweight models.

- Alternative Hypothesis (H_1): There is a significant difference in performance among the lightweight models.

The Friedman test is a non-parametric test used to determine if there are statistically significant differences among multiple groups. It is suitable for comparing multiple models across different metrics, as in this case. The significance level α for this analysis is set to 0.05. The Friedman test was implemented using Python with the 'scipy' library. The 5-fold accuracy results, for each model were collected and arranged into a data matrix. The 'friedmanchisquare' function from 'scipy.stats' was then used to calculate the Friedman statistic and corresponding p-value.

Interpreting the results involves considering the obtained p-value. If the p-value is less than the chosen significance level, it suggests that at least one of the models differs significantly from the others. Conversely, if the p-value is greater than the significance level, there's insufficient evidence to conclude that there are differences among the models. Therefore, the null hypothesis cannot be rejected.

Model Comparison using a Combined Metric Approach

If the Friedman test has rejected the null hypothesis, then we compare the models based on a Combined Metric Approach. This approach aims to provide a comprehensive evaluation of model performance in resource-constrained environments by integrating multiple evaluation criteria, including accuracy, inference time, and model size (number of parameters), into a single metric.

- **Metrics for Model Comparison:** Each metric—accuracy, inference time, and model size—is evaluated in the context of assessing lightweight models in resource-constrained environments. These metrics provide insight into distinct aspects of model performance, encompassing accuracy for predictive capability, inference time for computational efficiency, and model size for resource utilization. Integrating these metrics ensures a comprehensive evaluation that captures the multidimensional nature of model performance

- **Weighting of Metrics:** Equal weights are assigned to each metric in the combined metric approach. This weighting scheme reflects a balanced consideration of accuracy, inference time, and model size, acknowledging their relative importance in resource-constrained environments. By giving equal importance to each metric, we aim to provide a fair and unbiased assessment of model performance across different dimensions.

- **Holistic Evaluation of Model Performance:** The combined metric provides a holistic evaluation of model performance by synthesizing multiple criteria into a single metric. This approach allows us to compare lightweight models comprehensively, taking into account their accuracy, efficiency, and model size simultaneously. By considering these factors together, we gain insights into the overall suitability of models for deployment in resource-constrained settings.

A Python script was implemented to compute a combined metric for evaluating the performance of lightweight models across accuracy, inference time, and model size dimensions. The script first defines model data, including accuracy, inference time, and parameters, for each lightweight model under consideration. Subsequently, it normalizes the values of each metric to ensure comparability across models. Equal weights are assigned to the metrics, reflecting their equal importance in the evaluation process. The combined metric for each model is then calculated by aggregating the normalized values using the assigned weights. Finally, the models are ranked based on their combined metric values, providing insights into their relative performance across different dimensions. The code implementation enables a systematic and quantitative assessment of lightweight models, facilitating informed decision-making in resource-constrained environments where both performance and efficiency are critical factors.

4.2.2 Experimental Setup

Software Environment

Jupyter Notebook was selected as the coding platform which is based on the Python language. This choice offers a powerful and user-friendly environment for the experiments in developing and evaluating image classification models. Jupyter Notebooks enable an interactive development process and efficient execution, providing a robust platform for exploration. Their modular structure facilitates a step-by-step workflow, aiding in understanding complex tasks such as data preparation, model architecture definition, and training. With the seamless integration of code, visualizations, and documentation, Jupyter Notebooks make it easy to share and replicate our research [58]. The use of Python aligns well with TensorFlow and Keras libraries, forming a versatile ecosystem for deep learning experimentation [59]. The readability and expressiveness of Python code, coupled with extensive open-source contributions in the machine learning community, ensure access to state-of-the-art methodologies and functionalities. This combination of Jupyter Notebooks and Python not only simplifies the experimental pipeline but also promotes collaboration and knowledge dissemination within the research community.

In the experimentation, the following Python libraries were applied,

- **Numpy:** NumPy is a fundamental library in the realm of scientific computing using Python. Its strength lies in its ability to handle large, multi-dimensional arrays and matrices efficiently. NumPy also provides a comprehensive suite of mathematical functions, enabling straightforward operations on these arrays. This library is foundational for a wide range of numerical and computational tasks [60].

- **Pandas:** Pandas is an essential library for data manipulation and analysis in Python. It introduces a versatile data structure called DataFrame, which proves invaluable for handling structured data commonly encountered in machine learning tasks. With its powerful capabilities, Pandas simplifies tasks such as data cleaning, exploration, and preparation [61].

- **Tensorflow and Keras:** TensorFlow is a robust, open-source machine learning library that supports the development and training of a wide range of machine learning models, especially deep neural networks. Known for its flexibility and scalability, TensorFlow is favored for various AI applications [59]. Keras, a user-friendly layer on top of TensorFlow, simplifies neural network construction with its high-level abstractions, enabling a focus on model design over implementation details [62].

- **Pytorch and Torchvision** PyTorch is a dynamic, open-source machine learning library widely used for developing and training deep neural network models due to its intuitive design and flexibility. It is particularly favoured for academic research and prototype development [63]. Torchvision, an extension of PyTorch, provides popular datasets, model architectures, and common image transformations for computer vision applications, facilitating efficient image preprocessing and augmentation tasks [64].

- **Skikit-Learn:** Scikit-learn stands as a prominent machine learning library that provides a straightforward and efficient toolkit for data mining and analysis. Its offerings encompass an array of tools for tasks such as classification, regression, clustering, and model selection. With its user-friendly design, Scikit-learn facilitates the implementation and evaluation of machine learning algorithms, making it a favored choice for practitioners at various skill levels [59].

- **Matplotlib:** Matplotlib is a widely-used plotting library, offering diverse tools for creating static, interactive, and animated visualizations in Python. Its applications extend to various domains, and in the context of machine learning, Matplotlib is frequently employed to visualize training curves, confusion matrices, and other crucial metrics during the evaluation of models [65].

4.2.2.1 Hardware environment

The hardware used in this experiment is mentioned in Table 4.2.

CPU	Intel(R) Core(TM) i5-1135G7
GPU	Intel(R) Iris(R) Xe Graphics
Installed RAM	16GB
Operating System	Windows 11

Table 4.2: Hardware environment

4.2.3 Dataset

This dataset, with its diversity and authenticity, forms the cornerstone of this exploration into image classification methodologies within the realm of animal species.

- **Overview:** This dataset, consisting of 5,400 animal images across 90 diverse categories, serves as the foundation for our image classification study. Animals, fascinating members of the biological kingdom Animalia, exhibit a range of characteristics and interactions. From microscopic organisms to giants spanning 33.6 meters, animals play vital roles in shaping ecosystems.

- **Composition:** The dataset covers a wide spectrum of animal classes, reflecting the richness of the animal kingdom. It includes insects, vertebrates, and more, capturing the complexity of life forms.

- **Image Count:** With a total of 5,400 images and 90 different species each representing a unique specimen, the dataset provides a comprehensive and varied collection for training and evaluating image classification models.

- **Structure:** The Dataset has meticulously organized images to ensure a balanced representation across categories. This structure prevents biases during model training, enhancing the model's robustness and ability to generalize.

- **Data Source:** The dataset originates from Kaggle [66], a well-known data science platform. Kaggle, in collaboration with Google Images [67], curated the dataset. The use of Google Images enriches the dataset, offering a diverse array of authentic animal representations.

- **Acknowledgment:** We extend our gratitude to Kaggle for hosting the dataset and to Google Images for providing a valuable source of diverse images. This collaboration has made the dataset accessible for research and experimentation. Researchers and practitioners are invited to explore and utilize this dataset for advancing image classification techniques within the fascinating world of animal species.

Figure 4.3: Sample of Dataset images.

4.2.4 Implementation

Following an in-depth examination of existing literature, the adoption of lightweight architecture models takes center stage in this study. Notably, MobileNetV2, EfficientNetB3, ShuffleNet, SqueezeNet, and MnasNet have been chosen for their prowess in image classification tasks, each with its unique strengths. MobileNetV2 is particularly recognized for its suitability in resource-constrained scenarios, making it an ideal candidate for deployment on mobile devices and edge computing environments [14]. In contrast, EfficientNetB3 strikes a balance between accuracy and computational efficiency, positioning itself as a versatile choice for a variety of tasks [13]. ShuffleNet and SqueezeNet further extend the range of efficient architectures with their innovative use of pointwise group convolutions and fire modules, respectively, significantly reducing parameter counts and computational demands without sacrificing performance [6, 7]. MnasNet incorporates AutoML techniques to optimize its architecture for both high accuracy and efficiency on mobile devices [8]. The selection of these models is strategic, aiming to optimize computational efficiency without compromising the accurate classification of animal images.

Dataset Preparation

In the methodology of our experimentations, dataset preparation is a critical initial step applicable across all neural network models used in this study. This process ensures that the input data is in a suitable form for the learning algorithms, allowing for optimal performance and efficacy in model training.

Collection and Labeling Process: Datasets were systematically gathered from structured directories, where each sub-directory corresponds to a different class label. This organization facilitated the automated collection of images along with their corresponding labels, ensuring a comprehensive dataset that includes representative samples of all categories involved in the study. Each image file's path and its label were parsed and stored in structured data formats for subsequent processing.

Label Encoding and Numerical Transformation: For the models to process the categorical labels effectively, labels were converted into numerical formats using a Label Encoder. This encoding process assigns a unique integer to each label, transforming categorical data into a machine-readable form. The mathematical representation of this encoding process can be described as follows: Given a label set L where $L = \{l_1, l_2, \ldots, l_n\}$ represents all unique labels, the Label Encoder function $f : L \to \{0, 1, 2, \ldots, n-1\}$ maps each label l_i to a unique integer.

Resizing and Normalization: To ensure consistency in input data, all images were resized to uniform dimensions of 224x224 pixels. This standardization is crucial as it affects the neural network's ability to learn effectively from the data. Post-resizing, the images were normalized to scale the pixel values to a range of 0 to 1. Normalization is performed using the formula:

$$\text{Normalized Pixel Value} = \frac{\text{Original Pixel Value}}{255}$$

This scaling helps in reducing the variance in input values, leading to faster convergence during training.

Stratified Train-Test Split Procedure: The dataset was split into training, validation, and testing subsets using a stratified train-test split to maintain a uniform distribution of classes across each dataset. This method ensures that each split of the data accurately reflects the overall distribution of the data, crucial for the robustness and generalizability of the models. Typically, 80% of the data was allocated for training while the remaining 20% was used for testing. The stratification was controlled by a random seed to ensure reproducibility of the dataset splits.

This structured approach to dataset preparation lays the groundwork for effective model training, ensuring that all models operate under the same initial conditions, thereby providing a fair basis for comparing their performances.

Training Strategy

The training strategy for all neural network models in this study involves several key techniques designed to optimize performance and ensure robust model training. These strategies are common across the different architectures employed and are crucial in achieving high levels of accuracy and efficiency.

Adam Optimizer: The Adam optimizer, a stochastic optimization method, is utilized across all models due to its effectiveness in handling sparse gradients and adapting the learning rates for different parameters [68]. The key benefit of Adam is its combination of the advantages from two other extensions of stochastic gradient descent: Adaptive Gradient Algorithm (AdaGrad) and Root Mean Square Propagation (RMSProp). Adam calculates an exponential moving average of the gradient and the squared gradient, and the parameters beta1 and beta2 control the decay rates of these moving averages. The general update rule for Adam is given by:

$$\theta_{t+1} = \theta_t - \frac{\eta}{\sqrt{\hat{v}_t} + \epsilon}\hat{m}_t$$

where θ represents the parameters, η is the learning rate, $\hat{m}_t$ and $\hat{v}_t$ are bias-corrected estimates of the first and second moments of the gradients, respectively, and ϵ is a small scalar added for numerical stability.

Data Augmentation Techniques: Data augmentation plays a crucial role in preventing overfitting and enhancing the generalization ability of the models. Common techniques include random transformations such as rotation, shifting, scaling, and flipping. These augmentations simulate variations that could occur in real-world scenarios, thereby broadening the model's experience during training, which helps improve its ability to generalize from the training data to new, unseen data.

Regularization Techniques: Regularization techniques such as Dropout and Batch Normalization are implemented to further enhance model training. Dropout prevents overfitting by randomly setting a fraction of input units to zero at each update during the training phase. This method helps in creating a noise-resistant and robust network. Batch Normalization, on the other hand, normalizes the input layer by adjusting and scaling the activations, allowing for higher learning rates and reducing the sensitivity to the initial weights [69].

$$\text{Batch Normalized Output} = \gamma \left(\frac{x - \mu_B}{\sqrt{\sigma_B^2 + \epsilon}} \right) + \beta$$

where x is the input to a layer, μ_B and σ_B^2 are the mean and variance computed over the current batch, γ and β are parameters to be learned, and ϵ is a small constant for numerical stability.

Callbacks: Callbacks such as EarlyStopping and ReduceLROnPlateau are employed to optimize the training process. EarlyStopping monitors a specified metric, such as validation loss, and stops the training process if the metric stops improving, thus preventing overfitting and reducing computational waste. ReduceLROnPlateau reduces the learning rate when a metric has stopped improving, which helps in fine-tuning the model by taking smaller steps in the parameter space, thereby finding a better local minimum.

Model Evaluation

Model evaluation is a crucial phase in the training process, allowing for an assessment of a model's performance beyond its training data. This section outlines the standard methodologies and metrics used to evaluate the effectiveness of the neural network models employed in this study.

Performance Metrics: The primary metrics used to assess the models include accuracy, F1 score, precision, recall, and loss. Each metric offers unique insights into the model's performance:

- **Accuracy:** This metric measures the proportion of correct predictions among the total number of cases processed. It is particularly useful as a general indicator of model performance but can be misleading when dealing with imbalanced datasets. The formula for accuracy is given by:

$$\text{Accuracy} = \frac{\text{Number of Correct Predictions}}{\text{Total Number of Predictions}}$$

- **Precision:** Precision measures the accuracy of positive predictions. Formulated as the ratio of true positive predictions to the total predicted positives, precision is critical when the costs of false positives are high. The formula for precision is:

$$\text{Precision} = \frac{\text{True Positives}}{\text{True Positives} + \text{False Positives}}$$

- **Recall:** Also known as sensitivity, recall measures the ability of a model to find all the relevant cases within a dataset. It is defined as the ratio of true positives to the actual number of positives, which includes the false negatives. The formula for recall is:

$$\text{Recall} = \frac{\text{True Positives}}{\text{True Positives} + \text{False Negatives}}$$

- **F1 Score:** The F1 score is the harmonic mean of precision and recall, providing a balance between them. It is particularly valuable when the classes are imbalanced. F1 is calculated as:

$$\text{F1 Score} = 2 \times \frac{\text{Precision} \times \text{Recall}}{\text{Precision} + \text{Recall}}$$

- **Loss:** Often, the loss function used during training is also used to evaluate the model. For classification tasks, categorical cross-entropy loss is commonly used. It measures the difference between the predicted probability distributions and the true distribution across all classes. The formula for categorical cross-entropy loss is:

$$\text{Loss} = -\sum_{c=1}^{M} y_{o,c} \log(p_{o,c})$$

where M is the number of classes, $y_{o,c}$ is the binary indicator (0 or 1) if class label c is the correct classification for observation o, and $p_{o,c}$ is the predicted probability that observation o belongs to class c.

- **Average Inference Time per Batch:** This metric measures the average time taken by the model to process each batch of data during inference. It is crucial to assess the real-time performance of the model in deployment scenarios.

$$\text{Average Inference Time per Batch} = \frac{\text{Total Inference Time}}{\text{Number of Batches}}$$

- **Number of Parameters:** The number of parameters in a model indicates its complexity and capacity to learn from data. It is essential for understanding the model's size and computational requirements during training and inference. It can be obtained using the summary command on the model.

4.2.5 Model-Specific Implementations

MobileNetV2

Architecture:

MobileNetV2 is utilized in this study for its efficiency and effectiveness in handling image classification tasks, particularly suitable for environments with limited computational resources. The model architecture is leveraged from TensorFlow Hub and consists of a sequence of layers specifically optimized for mobile devices. It employs depthwise separable convolutions that significantly reduce the number of parameters without sacrificing model performance [14].

Unique Settings:

The model is initially set with its base layers frozen, meaning that the pre-trained weights are not updated during the first phase of training. This approach allows the model to retain the generic features it has learned from a vast and diverse image dataset (ImageNet). Specific settings for this model implementation include:

- Image size fixed at 224×224 pixels for both height and width.

- Base model layers are set to non-trainable to preserve learned features during initial training phases.

Training and Fine-Tuning:

Training involves several key strategies:

- **Data Augmentation:** Implemented via an image data generator, this includes random rotations, width and height shifts, shear intensity adjustments, zoom augmentations, and horizontal flips to enhance model robustness.
- **Early Stopping:** Utilized to halt training when validation accuracy does not improve for three consecutive epochs, helping prevent overfitting and ensuring optimal model performance.
- **Fine-Tuning:** After initial training, fine-tuning is performed by unfreezing the entire model while keeping the batch normalization layers non-trainable to refine feature extraction without destabilizing the learned weights. This is followed by continued training with a significantly reduced learning rate ($1e{-}5$), allowing subtle adjustments and improvements in model accuracy.

EfficientNetB3

In the development of the EfficientNet-based image classification model, portions of the code and methodology were adapted from Hamed Ghorbani's project on Kaggle, which demonstrated an implementation for classifying images across 90 different animal categories with a high accuracy rate. This resource provided foundational insights and code snippets which were integral to the setup and execution of the model in this research. For further details, see the original project at Kaggle: [70].

Architecture:

EfficientNetB3 was chosen for its compelling balance between computational efficiency and accuracy. This architecture scales uniformly at the level of depth, width, and resolution, which contributes significantly to improving model efficiency and performance on complex image classification tasks. Its architecture leverages a compound scaling method that optimizes these three dimensions through a fixed set of scaling coefficients, ensuring that each layer captures appropriate features at different scales and complexities.

Unique Settings:

For the EfficientNetB3, several unique settings were adjusted to tailor the model to the specific dataset and objectives of the project:

- **Mixed Precision Training:** Utilization of TensorFlow's mixed precision to expedite training, which decreases memory usage and increases the training speed without losing the model's accuracy or stability.
- **Batch Normalization:** Layers were selectively frozen during the initial training phases to stabilize the learning process by normalizing the inputs to each layer.

- **Progressive Resizing:** Employed as a part of data preprocessing, which adjusts the image resolutions during different phases of training, gradually increasing from smaller to target sizes, enhancing the model's ability to generalize across various image dimensions.

Training and Fine-Tuning:

Training of the EfficientNetB3 incorporated advanced techniques to maximize performance:

- **Advanced Augmentation Techniques:** Included random flips, rotations, zooms, and contrast adjustments. These techniques simulate different photographic conditions, thereby improving the robustness of the model against overfitting and enhancing its ability to generalize.

- **Fine-Tuning Strategy:** Post initial training, fine-tuning was conducted by unfreezing the entire model while keeping the batch normalization layers non-trainable to refine feature extraction without destabilizing the learned weights. The learning rate was significantly reduced to fine-tune nuanced features crucial for accurate classifications.

ShuffleNet

Architecture:

ShuffleNet is designed with efficiency in mind, particularly suited for mobile and embedded vision applications. It utilizes pointwise group convolutions and channel shuffle operations to reduce computational cost while maintaining comparable accuracy significantly [6]. The architecture is designed to maintain a balance between latency and accuracy, making it optimal for deployment in resource-constrained environments.

Unique Settings:

The unique configuration of the ShuffleNet in our study includes:

- **Group Convolutions:** These significantly reduce the number of parameters compared to regular convolutions, thus decreasing the computational load.

- **Channel Shuffling:** Ensures that cross-group information is adequately propagated, preventing bottlenecks that typically occur with group convolutions alone.

- **Pre-trained Model Utilization:** The model was initialized with weights pre-trained on ImageNet, focusing on transferring learned features to the new dataset, accelerating the initial learning phase.

- **Custom Data Loaders:** Custom dataset classes and data loaders were utilized to handle image data effectively, ensuring that data transformations and batching operations are optimized for training efficiency.

Training and Fine-Tuning:

Training of ShuffleNet involved specific strategies tailored to exploit its architectural efficiencies:

- **Transfer Learning:** The model was initialized with weights from a model pre-trained on ImageNet. Only the final fully connected layer was trained initially to adapt the pre-trained features to the new task, significantly reducing the training time. The new fully connected layer is defined by the following line of code:

  Listing 4.1: Replacing the final fully connected layer in ShuffleNet

  ```
  new_fc = nn.Linear(model.fc.in_features,
    ↪ num_classes)
  ```

- **Data Augmentation:** Extensive augmentation techniques such as random resized cropping, horizontal flipping, and random rotation were applied to the training images to improve generalization and robustness of the model.

- **Fine-Tuning:** After the initial phase, fine-tuning was performed by unfreezing all the layers of the model and continuing training under a reduced learning rate. This approach allowed fine adjustments to the model weights, enhancing its accuracy on the dataset.

SqueezeNet

Architecture:

SqueezeNet is renowned for its parameter efficiency, which makes it an excellent choice for applications where model size and speed are critical. Its architecture utilizes a design strategy that replaces standard convolutions with fire modules consisting of a squeeze convolution layer (which has only 1x1 filters) followed by an expanded layer that has a mix of 1x1 and 3x3 convolution filters [7]. This design significantly reduces the number of parameters without sacrificing accuracy, which justifies its use in scenarios demanding high computational efficiency.

Unique Settings:

During the setup for training SqueezeNet, several unique aspects were considered:

- **Custom Data Loaders:** Custom dataset classes and data loaders were utilized to handle image data effectively, ensuring that data transformations and batching operations are optimized for training efficiency.

- **Pre-trained Model Utilization:** The model was initialized with weights pre-trained on ImageNet, focusing on transferring learned features to the new dataset, accelerating the initial learning phase.

Training and Fine-Tuning:

The training and fine-tuning of SqueezeNet were particularly focused on the newly added classifier layer:

- **Classifier Adaptation:** The final classifier layer of the model was replaced with a new convolutional layer tailored to the number of unique labels in the dataset. This layer is defined in the model as follows:

Listing 4.2: Modifying the classifier layer in SqueezeNet

```
model.classifier[1] = nn.Conv2d(512, len(set(df['
    ↪ labels'])), kernel_size=(1, 1))
```

- **Transfer Learning Approach:** Initially, all layers except the newly added classifier were frozen. This approach focuses the training on adapting the model to the specific output classes of the current dataset.

- **Fine-Tuning:** After the initial training phase, a fine-tuning process was conducted by unfreezing all layers. The learning rate was significantly reduced to allow subtle adjustments to the weights, improving the accuracy without large deviations from the learned features.

MnasNet

Architecture:

MnasNet was specifically selected for its superior balance of accuracy and computational efficiency, making it highly suitable for deployment on mobile devices. MnasNet employs an automated machine learning (AutoML) approach for neural architecture search, focusing on optimizing both accuracy and latency. The architecture is designed to be lightweight and fast, with separable convolutions that reduce the number of parameters and computational cost without compromising the model's performance [8].

Unique Settings:

Several optimizations were tailored specifically for MnasNet to enhance its performance and efficiency:

- **Pre-trained Weights:** MnasNet was initialized with weights pre-trained on ImageNet, providing a solid foundation of learned features and significantly speeding up the convergence during training.

- **Custom Data Loaders:** Custom dataset classes and data loaders were utilized to handle image data effectively, ensuring that data transformations and batching operations are optimized for training efficiency.

- **Device Optimization:** The entire model was adapted to operate seamlessly on GPU if available, utilizing CUDA for accelerated computations. This adaptation ensures optimal performance during both training and inference phases.

Training and Fine-Tuning:

The training and fine-tuning of MnasNet were strategically focused on both leveraging pre-trained capabilities and optimizing the network for specific tasks:

- **Classifier Adaptation:** Similar to the approach with SqueezeNet, the final classifier layer of MnasNet was replaced with a new layer customized to the number of unique classes in the dataset. This modification is essential for tailoring the network's output to the specific classification task at hand.

Listing 4.3: Modifying the classifier layer in MnasNet

```
model.classifier = torch.nn.Linear(model.
    ↪ classifier.in_features, num_classes)
```

- **Transfer Learning Approach:** Initially, all parameters of the model, except the newly added classifier, were frozen. This selective training focuses the learning process on the new classifier, ensuring that the network adapts effectively to the new labels without altering the pre-trained features.

- **Fine-Tuning:** After the initial training phase, the fine-tuning phase involved unfreezing all layers of the network. The learning rate was substantially lowered to make fine adjustments to the network weights, which enhances the model's accuracy and ability to generalize from pre-trained features to specific tasks.

Validating the Model

Validation is a crucial step in the training process to assess how well a model generalizes to new, unseen data. It helps in detecting overfitting and ensures that the model is learning patterns that apply to various instances. Cross-validation is a robust validation technique that further enhances model evaluation by providing a more comprehensive assessment.

5-Fold Cross-Validation in the Experiment:

In this study, 5-fold cross-validation was employed for model evaluation. The dataset was divided into five subsets, and the model was trained and evaluated five times, each time using a different subset as the validation set while the remaining subsets were used for training. This method ensures that every instance in the dataset has an opportunity to be used in both training and validation.

The performance metrics, including accuracy, precision, recall, and F1 score, were calculated for each fold and then averaged to provide a comprehensive and reliable assessment of the model's effectiveness. This approach reduces the likelihood of performance being affected by a particular data split and gives a more accurate picture of the model's generalizability.

This robust evaluation strategy ensures that the reported model performance is representative across various data distributions and contributes to the reliability and validity of the study's findings.

Chapter 5

Results and Analysis

5.1 Literature Review Results

After thoroughly examining the existing works of literature in Section 4.1, our research has identified effective lightweight models for further experimentation. Our review revealed several algorithms that exhibit promise in accurately classifying images. In Table 5.1, we present a condensed overview of the outcomes derived from our literature investigation, concentrating on the algorithms earmarked for potential utilization in our research. The column labeled "Author's Name" specifies the authors of the respective research papers. The "Algorithms Used" column outlines the specific algorithms implemented in each study. The "Algorithms Considered" column lists all the algorithms that were considered for the experimentation. The "Reason for Selection" column provides the rationale behind choosing the specific algorithms that were ultimately used in the research.

5.1.1 Results Table

Author's Name	**Algorithms Used**	**Algorithms Considered**	**Reason for Selection**
V. A. O. Nancy et al [43]	MobileNetV2, CNN model with 3 or 5 layers, KNN, SVM, RF and Others	MobileNetV2	High accuracy of 97.58% with fine-tuning and balanced augmentation
Ammar Chalifah et al [48]	MobileNetV2, ResNet50V2, ResNet 152 V2, Xception, VGG16, and DenseNet12	MobileNetV2	Showed superior performance with high accuracy (91.28%) for COVID-19 classification.
Yonis Gulzar et al [45]	ALexNet, VGG16, InceptionV3 and MobileNetV3 and EfficientNetB3	EfficientNetB3	Outperformed other models with the highest accuracy (0.979) in disease classification.
Palla Reshma et al [46]	VGG19, ResNet50, and EfficientNetB3	EfficientNetB3	Displayed superior accuracy(92.4%) and efficiency for disease identification.

Author's Name	Algorithms Used	Algorithms Considered	Reason for Selection
Rajdeep Chatterjee et al [47]	VGG16, VGG19, ResNet50, MobileNetV1, and EfficientNetB3	EfficientNetB3	Outperformed other models, achieving the best-reported accuracies (88.78%) for pneumonia classification.
Jong Su Byun et al [44]	MobileNetV2, Xception, and InceptionV3	MobileNetV2	Showcased superior accuracy (98.57%) in lesion classification.
G. Li et al [49]	EfficientNetB3, MobileNetV2, ShuffleNet, SqueezeNet, MmasNet	Squeezenet	SqueezeNet performed well with 95.9% accuracy.
D. Avola et al [51]	MobileNetV2, ShuffleNet, SqueezeNet, MmasNet	MobileNetV2	MobileNetV2 demonstrated excellent performance, with an 83.78% accuracy rate.
T. Feng et al [50]	EfficientNet, MobileNetV2, ShuffleNet	ShuffleNet	With an accuracy percentage of 98.2%, the ShuffleNet model performs better.
M. Tan et al [8]	MmasNet, MobileNetV2	MmasNet	MnasNet achieves 76.7% accuracy.
K. Saddami et al [53]	MobileNetV2, ShuffleNet, SqueezeNet, MmasNet	Shuffle, SqueezeNet	All accurately determined 100%. SqueezeNet and ShuffleNet are ideal for systems with limited memory and computing power.
B. Syamsuri et al [54]	MmasNet, MobileNetV2	MmasNet	MnasNet achieved more higher accuracy 94.87% than MobileNet.
M. E. Karar et al [55]	ShuffleNet, MmasNet, MobileNetV2	MnasNet	MnasNet's accuracy of 99.00% surpassed other models, demonstrating balance accuracy and speed.
A. K. Hrithik et al [56]	SqueezeNet, ShuffleNet	ShuffleNet	ShuffleNet achieved a noteworthy 99.96% accuracy rate.
R. Sapkota et al [57]	ShuffleNet, SqueezeNet, MobileNetV2	SqueezeNet	SqueezeNet's 21.81 MB diminutive size was commended for achieving 100% accuracy.
V. Shwetha et al [52]	SqueezeNet	SqueezeNet	SqueezeNet was employed as a classifier, resulting in an amazing 97% total accuracy.

Table 5.1: Literature review results

5.1.2 Literature Review Analysis

Based on our thorough review of existing research papers, we've identified several promising lightweight models. The models considered, namely MobileNetV2, EfficientNetB3, ShuffleNet, SqueezeNet, and MnasNet, have all demonstrated superior performance in various studies, as highlighted in Table 5.1.

In the examination of multiple deep learning architectures for diverse classification tasks, each of these models has shown remarkable accuracy and efficiency, making them all strong candidates for our research. MobileNetV2 and EfficientNetB3 have consistently stood out for their exceptional performance. MobileNetV2 is highlighted for its efficiency and commendable accuracy, performing exceptionally well in various image recognition tasks, in COVID-19 detection from X-ray images [48]. EfficientNetB3, renowned for its transfer learning capabilities, was frequently selected for its ability to adapt learned features effectively to new tasks, achieving peak accuracy and showcasing robustness across different datasets, such as ocular disease identification [46].

ShuffleNet has also emerged as a significant contender, particularly recognized for achieving remarkable accuracy, such as a noteworthy 99.96% [56] in certain tasks. Its lightweight architecture makes it highly suitable for efficient image classification, especially in resource-constrained environments. Similarly, SqueezeNet has proven ideal for systems with limited memory and computing power, achieving high accuracy with a very small model size [52]. This makes it an excellent choice for applications where resource efficiency is critical. MnasNet, designed with a focus on mobile and edge devices, has shown competitive accuracy and efficiency with mobilenetv2 [54] [55], further enhancing the pool of models suitable for our research. Its design prioritizes both performance and resource constraints, making it a versatile model for various image classification tasks.

The selection of MobileNetV2, EfficientNetB3, ShuffleNet, SqueezeNet, and MnasNet is based on their proven effectiveness and efficiency in various studies. These models collectively offer a robust framework for developing high-accuracy, resource-efficient animal classification models. Employing these state-of-the-art techniques is expected to significantly enhance the precision and reliability of our predictions, thereby contributing valuable insights resource resource-constraint environments.

In conclusion, the recurrent selection of these architectures across multiple studies reinforces their status as go-to models for image classification tasks. Each model, with its unique strengths in accuracy, efficiency, and adaptability, plays a crucial role in the evolving landscape of deep learning applications in image recognition and analysis. Their collective use in our research is anticipated to yield precise and dependable results, showcasing their prowess in enhancing our animal classification models.

5.2 Experiment Results

This section serves as the core of this study, unveiling insights gained from the experimental phase. Through rigorous experimentation with different lightweight transfer learning models in the classification of animal images, this section delves into the performance metricsand the nuances of each approach. The findings provide a comprehensive understanding of how these models fare in real-world scenarios, shedding light on their strengths and limitations. This analysis is instrumental in drawing meaningful conclusions, informing future research directions, and offering practical implications for animal image classification.

Following the comprehensive implementation and 5-fold cross-validation of various lightweight transfer learning models featuring distinct architectures, this study unveils compelling insights into animal image classification. The models' efficacy is meticulously assessed using key performance metrics, including accuracy, F1 score, recall, and precision, derived from the cross-validated results. The results not only validate the selected models' effectiveness but also furnish valuable insights for their optimal utilization in real-world scenarios, addressing challenges posed by different levels of resource-constraint environments.

5.2.1 Results

The implemented transfer learning models, utilizing the MobileNetV2, EfficientNetB3, ShuffleNet, SqueezeNet, and MnasNet architecture, demonstrated commendable performance in classifying animal images. The training process involved two significant phases: initial training and fine-tuning.

MobileNetV2

Initial Training:

- The training accuracy consistently increased with each epoch, reaching 94.65% by Epoch 8.
- Validation accuracy followed a similar trend, peaking at 83.15% in Epoch 8.
- Both training and validation losses decreased, indicating that the model was effectively learning and generalizing.
- The relatively small gap between training and validation accuracy suggests good generalization.
- Validation accuracy stabilized towards the later epochs, indicating that further training might yield diminishing returns.

In summary, the model demonstrated effective learning on the training data with reasonable generalization to the validation set.

Fine Tuning:

- The initial model achieved strong performance with an accuracy of 83.70% on the validation set.
- Fine-tuning resulted in a slight decrease in performance, with test accuracy reaching 82.59%.
- The model maintained effective learning and generalization capabilities, despite the marginal decrease in accuracy during fine-tuning.
- It's important to consider the trade-off between model complexity and overfitting during fine-tuning. In this case, the fine-tuning process slightly reduced overfitting, as indicated by the changes in accuracy compared to the initial training phase.

In summary, while fine-tuning led to a minor decrease in validation accuracy, the model continued to demonstrate strong learning and generalization capabilities.

EfficientNetB3

Initial Training:

- The training accuracy increased steadily with each epoch, reaching 93.41% by Epoch 10.
- Validation accuracy also showed consistent improvement, peaking at 90.92% in Epoch 10.
- Both training and validation losses decreased, indicating effective learning and generalization.
- The model demonstrated good generalization, as the validation accuracy closely tracked the training accuracy.
- The model might benefit from additional epochs, as both training and validation accuracies were still improving by the end of Epoch 10.

In summary, the model exhibited effective learning and generalization, and the close tracking of training and validation accuracies suggests good model performance.

Fine Tuning:

- The fine-tuning phase improved the model's performance from the initial stages.
- Both training and validation accuracies showed a steady increase over epochs.
- The validation accuracy consistently tracked the training accuracy, indicating good generalization and learning without overfitting.
- Minor decreases in validation accuracy in some epochs suggested fluctuations in model performance.

- The model's final validation accuracy reached 92.77%, indicating strong performance after fine-tuning.

In summary, the fine-tuning process led to improvements in both training and validation accuracies, demonstrating that further training helped the model learn more nuanced patterns in the data. The validation accuracy maintained a consistent track with the training accuracy, suggesting a well-generalized model without significant overfitting.

ShuffleNet

Initial Training:

- Training Accuracy increases gradually over epochs, reaching 71.44% by Epoch 50.
- Validation Accuracy increases steadily, reaching 80.65% by Epoch 50.
- Both training and validation losses decrease consistently, indicating effective learning and generalization.
- Gap between Training and Validation Accuracy gradually decreases over epochs, suggesting improving generalization.

The ShuffleNet model demonstrates effective learning on the training data, with gradual improvements in validation accuracy indicating reasonable generalization. The gap between training and validation accuracy also reduces over epochs, indicating improving generalization.

Fine Tuning:

- Training Accuracy increases steadily over epochs, reaching 79.88% by Epoch 5.
- Validation Accuracy shows improvement over epochs, reaching 85.37% by Epoch 5.
- Both training and validation losses decrease consistently, indicating effective fine-tuning and generalization.
- Gap between Training and Validation Accuracy remains relatively small, suggesting reasonable generalization.
- Validation accuracy continues to improve over epochs, indicating that the model is still benefiting from fine-tuning.

The fine-tuning process has led to improvements in both training and validation accuracy, indicating that the model is effectively learning the specific features of the new dataset. The gap between training and validation accuracy remains small, suggesting good generalization.

SqueezeNet

Initial Training:

- Training Accuracy increases steadily over epochs, reaching 99.81% by Epoch 15.
- Validation Accuracy increases gradually, reaching 82.31% by Epoch 15.
- Both training and validation losses decrease consistently, indicating effective learning and generalization.
- Gap between Training and Validation Accuracy is present in initial training indicating some overfitting even after considering measures against it.
- Validation accuracy seems to stabilize towards the later epochs, indicating that further training may have diminishing returns.

Overall, SqueezeNet's initial training shows promising results with high training accuracy and reasonable validation accuracy. However, the presence of a gap between training and validation accuracy suggests potential overfitting. Further training beyond a certain point may yield diminishing returns, as indicated by the stabilization of validation accuracy towards the later epochs.

Fine Tuning:

- Both initial training and fine-tuning achieve high training accuracies (99.81% vs. 99.19%).
- The validation accuracy after fine-tuning (78.89%) is significantly lower than the validation accuracy after initial training (82.31%).

In summary, while both initial training and fine-tuning processes exhibited effective learning, fine-tuning did not lead to an improvement in validation accuracy, indicating potential limitations in adapting the model to the new dataset.

MnasNet

Initial Training:

- Training Accuracy Gradually increases over epochs, reaching 99.79% by Epoch 10.
- Validation Accuracy Shows a similar trend, reaching 89.54% by Epoch 10.
- Gap between Training and validation Accuracy Gradually increases over epochs but remains acceptable, indicating reasonable generalization.
- Validation accuracy seems to stabilize towards the later epochs, indicating diminishing returns from further training.

In summary, the model demonstrates effective learning on the training data, with both training and validation accuracies increasing gradually over epochs. While there is a slight increase in the gap between training and validation accuracy indicating reasonable generalization. The stabilization of validation accuracy towards the later epochs suggests that further training may have diminishing returns.

Fine Tuning:

- Both initial training and fine-tuning achieve high training accuracies (99.79% vs. 99.95%).
- Fine-tuning leads to a slight increase in validation accuracy (89.54% vs. 90.93%).
- A small gap between Training and Validation Accuracy is present in fine-tuning indicating slight overfitting.
- Validation accuracy shows some fluctuations but generally maintains a high level in both initial training and fine-tuning.

In summary, both initial training and fine-tuning processes achieve high accuracies, with fine-tuning resulting in a slight improvement in validation accuracy. However, there is a small gap between training and validation accuracy in fine-tuning, suggesting some degree of overfitting. Despite fluctuations, validation accuracy remains consistently high in both training and fine-tuning.

5-Fold Cross-Validation

To validate the above-obtained results, a 5-fold cross-validation was conducted on the models. The average values for Accuracy, Precision, Recall, and F1 Score were calculated and are presented in Table 5.2.

Model	Accuracy	Precision	Recall	F1
MobileNetV2	0.845	0.854	0.845	0.843
EfficientNetB3	0.924	0.930	0.919	0.914
ShuffleNet	0.879	0.881	0.882	0.876
SqueezeNet	0.806	0.826	0.806	0.806
MnasNet	0.897	0.898	0.898	0.892

Table 5.2: Averages of metrics for 5-Fold Cross-Validation Results

Accuracy

In Table 5.2, the accuracy of several models in classifying animal images is presented and recorded as follows: MobileNetV2 achieves an accuracy of 0.845, EfficientNetB3 achieves 0.924, ShuffleNet reaches 0.879, SqueezeNet scores 0.806, and MnasNet attains 0.897. This table offers a comprehensive insight into the performance of various algorithms in this classification task. The accuracy comparison results are graphically depicted in Figure 5.1.

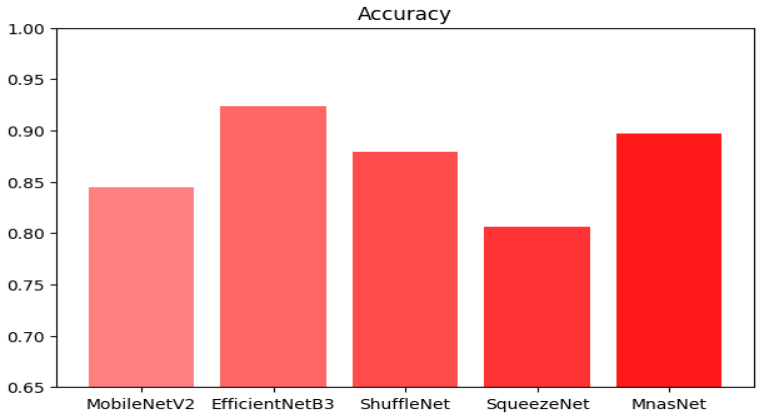

Figure 5.1: Comparison of accuracies

Precision

In Table 5.2, the precision of several models in classifying animal images is presented and recorded as follows: MobileNetV2 achieves a precision of 0.854, EfficientNetB3 achieves 0.930, ShuffleNet reaches 0.881, SqueezeNet scores 0.826, and MnasNet attains 0.898. This table offers a comprehensive insight into the performance of various algorithms in this classification task, now considering precision as an additional metric. The precision comparison results are graphically depicted in Figure 5.2.

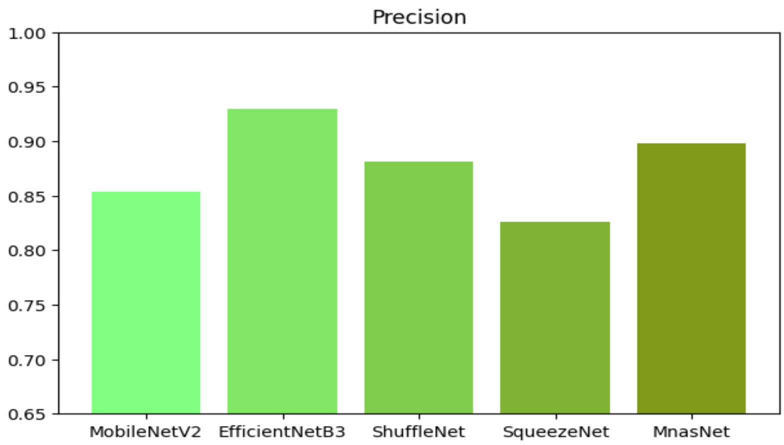

Figure 5.2: Comparision of precision

Recall

In Table 5.2, the recall of several models in classifying animal images is presented and recorded as follows: MobileNetV2 achieves a recall of 0.845, EfficientNetB3 achieves 0.919, ShuffleNet reaches 0.882, SqueezeNet scores 0.806, and MnasNet attains 0.898. This table offers a comprehensive insight into the performance of various algorithms in this classification task, now considering recall as an additional metric. The recall comparison results are graphically depicted in Figure 5.3.

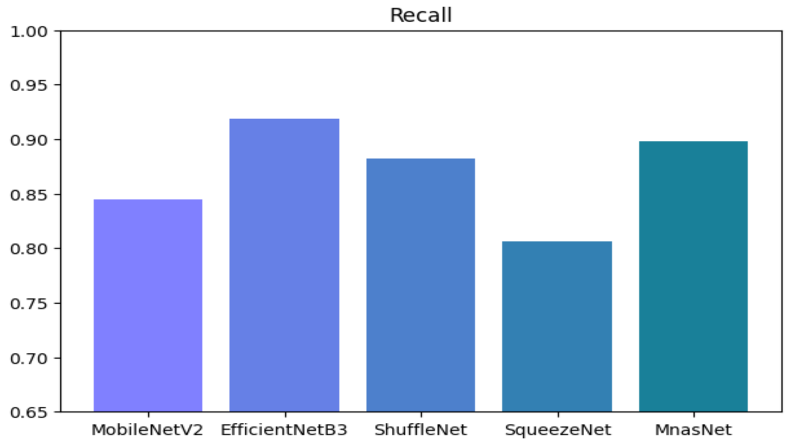

Figure 5.3: Comparision of recall

F1-Score

In Table 5.2, the F1 score of several models in classifying animal images is presented and recorded as follows: MobileNetV2 achieves an F1 score of 0.843, EfficientNetB3 achieves 0.914, ShuffleNet reaches 0.876, SqueezeNet scores 0.806, and MnasNet attains 0.892. This table offers a comprehensive insight into the performance of various algorithms in this classification task, considering the F1 score as an additional evaluation metric. The F1 score comparison results are graphically depicted in Figure 5.4

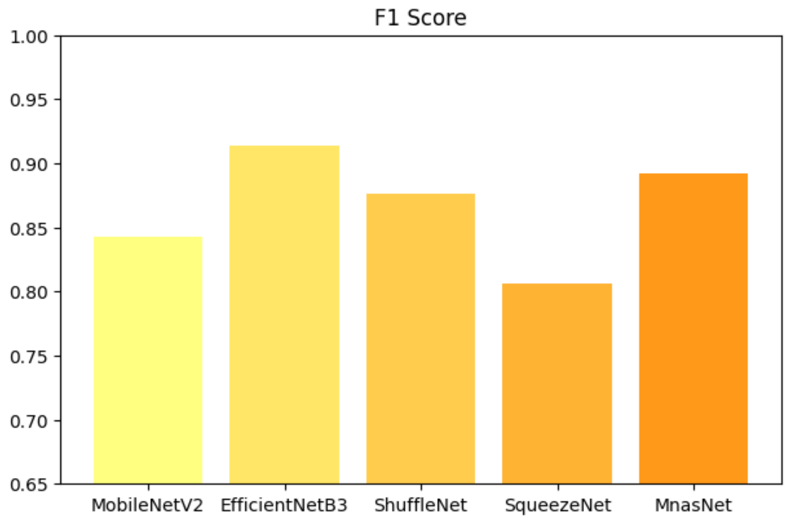

Figure 5.4: Comparision of F1-scores

5.2.2 Friedman Test Result

The Friedman Test was conducted to compare the accuracy of the models. The null hypothesis (H_0) stating no significant difference in accuracy between the models was rejected, providing evidence for a substantial distinction.

Friedman Statistic and P-Value:

- Friedman-Statistic: 20.00
- P-Value: 0.0005

Critical Values and Interpretation

The critical Friedman statistic value for the test with 5 models, 5 folds and $\alpha = 0.05$ is approximately 8.96 [71]. The calculated Friedman statistic $|20.00|$ far exceeds the critical value, reinforcing the rejection of the null hypothesis. The calculated p-value of 0.0005 is much smaller than the chosen significance level ($\alpha = 0.05$). Therefore, we reject the null hypothesis.

In hypothesis testing, when the p-value is below the significance level, it indicates that the observed results are unlikely to have occurred by random chance alone. As a result, we have strong evidence to support the alternative hypothesis, suggesting a significant difference in the context of the Friedman Test.

Conclusion

The results of the Friedman test provide robust evidence of a statistically significant difference in accuracy between all the models, by this we can confidently reject the null hypothesis.

Combined Metric Approach Results

The lightweight models—MobileNetV2, EfficientNetB3, ShuffleNet, SqueezeNet, and MnasNet—were ranked using a combined metric approach with equal weights assigned to accuracy, inference time, and model size. The initial metrics for each model, including accuracy, inference time (in seconds), and parameters (in millions), are summarized in the table 5.3:

Model	Accuracy	Inference Time (in Seconds)	Parameters (in Millions)
MobileNetV2	0.845	0.66	3.68
EfficientNetB3	0.924	1.38	11.2
ShuffleNet	0.879	0.29	1.35
SqueezeNet	0.806	1.13	0.77
MnasNet	0.897	0.63	3.2

Table 5.3: Model Size and Performance

Based on the combined metric scores calculated using equal weights for all initial metrics, the models were ranked as follows:

- ShuffleNet: Achieved the highest combined metric score of 0.854, indicating superior overall performance across accuracy, inference time, and model size dimensions.
- MnasNet: Ranked second with a combined metric score of 0.742, demonstrating strong performance across all evaluated metrics, albeit slightly lower than ShuffleNet.
- MobileNetV2: Positioned third, with a combined metric score of 0.571, indicating respectable performance but trailing behind ShuffleNet and MnasNet.
- SqueezeNet: Placed fourth with a combined metric score of 0.410, suggesting potential trade-offs in accuracy, inference time, or model size compared to the top-ranked models.
- EfficientNetB3: Ranked last with the lowest combined metric score of 0.333, indicating relatively weaker overall performance across the evaluated dimensions.

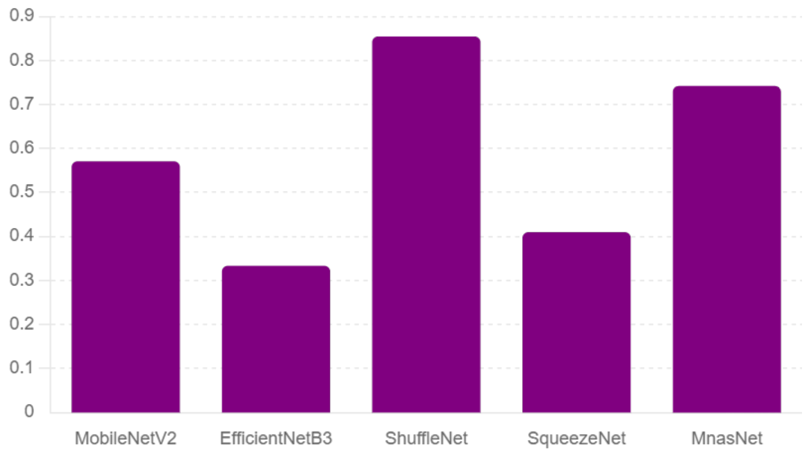

Figure 5.5: Model Combined Metric graph

Analysis of the Results:

The rankings based on the combined metric approach reveal interesting insights into the performance of the lightweight models:

- **ShuffleNet (Rank 1):** ShuffleNet achieved the highest combined metric score, primarily due to its competitive accuracy, low inference time, and moderate model size. Its efficient design, characterized by lightweight operations and network shuffling, enables fast inference while maintaining accuracy, making it well-suited for resource-constrained environments.

- **MnasNet (Rank 2):** MnasNet demonstrated strong performance across all evaluated metrics, particularly in accuracy and inference time. Its architecture, optimized using neural architecture search (NAS), strikes a balance between model complexity and performance, contributing to its high ranking.

- **MobileNetV2 (Rank 3):** MobileNetV2 exhibited respectable accuracy but lagged behind in inference time and model size compared to the top-ranked models. While its depthwise separable convolutions reduce model size, they may result in slightly higher inference times, impacting its overall ranking.

- **SqueezeNet (Rank 4):** SqueezeNet prioritizes model size reduction but may incur trade-offs in accuracy and inference time. Its compact architecture, characterized by fire modules and aggressive downsampling, leads to lower accuracy and longer inference times compared to higher-ranked models.

- **EfficientNetB3 (Rank 5):** EfficientNetB3 achieved the lowest combined metric score, primarily due to its larger model size and higher inference time. Despite its superior accuracy, the model's increased complexity and computational requirements result in reduced efficiency, leading to its lower ranking.

These analyses highlight the trade-offs involved in optimizing lightweight models for performance, efficiency, and model complexity. Each model's architecture and design choices influence its performance across different dimensions, underscoring the importance of considering multiple metrics in evaluating model suitability for resource-constrained environments.

Similar trends were observed when varying the weight scenarios. Assigning more weight to accuracy, inference time, or model size individually led to minor changes in the rankings, but the overall performance patterns remained consistent. ShuffleNet and MnasNet consistently performed well across different weight scenarios, confirming their robust balance of accuracy, efficiency, and model complexity.

Model Selection Guidelines

When selecting a model based on application constraints, ShuffleNet emerges as the best choice for scenarios with extremely limited resources, such as battery-powered wildlife cameras or mobile devices with limited processing and memory. ShuffleNet achieved the highest combined metric score of 0.854, balancing competitive accuracy with low inference time and moderate model size.

MnasNet, with a combined metric score of 0.742, is another excellent option, offering strong performance across all metrics and is suitable for environments with moderate constraints. MobileNetV2, with a combined metric score of 0.571, provides respectable performance and is ideal for applications requiring a balance between accuracy and resource usage. It is particularly useful in scenarios like automated species identification from remote camera trap images, where moderate accuracy is acceptable.

SqueezeNet, which achieved a combined metric score of 0.410, excels in scenarios where model size is a critical factor, such as when storage space is at a premium.

However, it may incur trade-offs in accuracy and inference time compared to higher-ranked models, making it suitable for applications where minimizing model size is more important than achieving the highest accuracy.

EfficientNetB3 demonstrated the highest validation accuracy of 92.77% but ranked lower in the combined metric due to its higher resource demands, with a combined metric score of 0.333. This makes it suitable for scenarios where classification reliability is critical, and there is adequate processing power, such as fast, efficient training and inference on continuous data streams for time-sensitive tasks.

Overall, the choice of model should be guided by the specific constraints and requirements of the application. ShuffleNet and MnasNet are versatile options for a range of resource-constrained environments, while MobileNetV2 and SqueezeNet offer tailored benefits for specific needs. EfficientNetB3, despite its higher resource demands, remains an excellent choice for high-accuracy tasks where processing power is not a limiting factor.

Chapter 6

Discussion

6.1 Reflection on the Results

RQ1: Which lightweight transfer learning architectures are most effective for classifying animals within the domain of image classification?

Answering to RQ1:

In response to Research Question 1 (RQ1), which seeks to identify effective lightweight transfer learning architectures for classifying animals within the domain of image classification, we conducted a comprehensive literature review as detailed in Section 4.1. Our exploration involved a thorough examination of existing studies and methodologies aimed at developing predictive lightweight models for animal image classification.

The literature review highlighted that while many lightweight models have been widely used for general image classification tasks, their application specifically to animal image classification, especially across a diverse range of species, has been relatively limited. This gap posed a challenge in directly determining the most effective models for our specific needs. Consequently, our research focused on identifying and assessing novel and efficient transfer learning architectures that demonstrate strong performance in various domains and have the potential to excel in animal classification tasks in resource-constraint environments.

From our investigation, MobileNetV2, EfficientNetB3, ShuffleNet, SqueezeNet, and MnasNet emerged as the most promising architectures. These models were selected based on their documented high accuracy, efficiency, and suitability for resource-constrained environments, as summarized in Table 5.1. Each of these models has shown exceptional performance in different image classification tasks:

- MobileNetV2: Noted for its high accuracy and efficiency, achieving up to 97.58% accuracy in study [43], outperforming other state-of-the-art image classification models in this research. It is particularly suitable for applications requiring both speed and accuracy.

- EfficientNetB3: Frequently highlighted for its superior performance, achieving better accuracy than its competition in these comparison studies [46] [47]. Its high accuracy and efficiency make it a strong candidate for various classification tasks.

- ShuffleNet: Recognized for its lightweight architecture and high accuracy, achieving superior results as demonstrated in study [50], even with limited computational resources.
- SqueezeNet: Praised for its compact size and efficiency, making it ideal for systems with constrained memory and processing power. Despite its small size, it achieves high accuracy rates and outperforms other lightweight models in several studies [49] [57].
- MnasNet: Known for its balanced accuracy and speed, designed specifically for mobile applications. It achieved accuracy rates surpassing 94.87% in some cases, even outperforming MobileNetV2 [54]. This makes it a viable option for mobile applications where both accuracy and speed are critical.

These models were chosen not only for their high performance but also for their efficient nature. Our focus on these cutting-edge techniques stems from their advanced capabilities, which align well with the requirements of accurately classifying a wide variety of animal species. Therefore, in answering RQ1, we have identified MobileNetV2, EfficientNetB3, ShuffleNet, SqueezeNet, and MnasNet as the most effective lightweight transfer learning architectures for classifying animals. Their proven performance in various contexts provides a solid foundation for their application in our specific task of animal image classification, paving the way for further experimentation and validation.

RQ2: How accurate are models trained using lightweight transfer learning architectures in animal image classification, and what practical implications do they hold?

Answering to RQ2:

This research question aims to assess the performance of selected lightweight transfer learning models for animal image classification in resource-constrained environments. Using a balanced dataset of 5,400 images across 90 animal categories from Kaggle and Google Images, models like MobileNetV2, EfficientNetB3, ShuffleNet, SqueezeNet, and MnasNet were chosen for their efficiency and accuracy. Data preparation included systematic collection, labeling, resizing to 224x224 pixels, normalization, and a stratified train-test split. The training process utilized the Adam optimizer, data augmentation, regularization techniques, and callbacks to prevent overfitting. Evaluation metrics such as accuracy, precision, recall, F1 score, loss, inference time, and parameter count provided a comprehensive performance assessment. The Friedman test and a combined metric approach were used to rank the models based on accuracy, inference time, and model size. This rigorous methodology ensures robust training, validation, and model comparison.

Following this methodology, the results demonstrated that MobileNetV2 achieved a validation accuracy of 83.15% initially, which slightly decreased to 82.59% after fine-tuning. EfficientNetB3 showed strong performance with validation accuracy improving from 90.92% to 92.77%. ShuffleNet displayed steady improvement, reaching validation accuracy from 80.65% to 85.37% after fine-tuning. SqueezeNet had high training accuracy but lower validation accuracy indicating potential overfitting, with validation accuracy decreasing from 82.31% to 78.89% after fine-tuning. MnasNet achieved consistent performance, with validation accuracy improving from 89.54% to 90.93%. The Friedman test confirmed significant differences in model performance (Friedman statistic: 20.00, p-value: 0.0005), leading to a combined metric approach that integrated accuracy, inference time, and model size. ShuffleNet ranked highest with a combined metric score of 0.854, followed by MnasNet (0.742) and MobileNetV2 (0.571), while SqueezeNet and EfficientNetB3 ranked lower with scores of 0.410 and 0.333, respectively. This analysis, showing trade-offs between metrics, guides the selection of optimal models for resource-constrained environments and informs future research and practical deployment strategies in animal image classification.

The practical implications of the evaluated lightweight transfer learning models are significant for resource-constrained environments. ShuffleNet, with the highest combined metric score of 0.854, is ideal for mobile and embedded systems in wildlife monitoring and agriculture due to its balance of accuracy, low inference time, and small model size. MnasNet also shows strong potential for similar applications, ranking second with a score of 0.742. MobileNetV2, while slightly lower in performance, remains a viable option with its respectable accuracy and moderate resource requirements. Although SqueezeNet and EfficientNetB3 scored lower, they provide valuable insights for specific use cases. These findings guide the selection and deployment of optimal models, ensuring effective and efficient animal image classification in practical, resource-constrained applications.

In conclusion, the practical implications of this study underscore the viability of utilizing lightweight transfer learning models like ShuffleNet and MnasNet in real-world applications requiring efficient and accurate animal image classification. MobileNetV2 also remains a valuable tool for moderately constrained environments. These models' reliability, generalizability, and adaptability position them as significant contributors to advancements in wildlife monitoring, and conservation efforts.

6.2 Threats to validity

6.2.1 Internal Validity

By standardizing the data preprocessing and model evaluation processes, we took measures to enhance internal validity when addressing Research Question 2 (RQ2). This study maintains its internal validity by using the same methods for all models, including MobileNetV2, EfficientNetB3, ShuffleNet, SqueezeNet, and MnasNet. Keeping the data preparation, division, and improvement consistent, ensures that any differences seen in model performance are due to the models themselves and not other factors. Additionally, using a robust test method like 5-fold cross-validation helps ensure that the models are evaluated consistently across different parts of the dataset, making the results more reliable and consistent.

6.2.2 External Validity

Our study goes beyond animal image classification because the models we selected, MobileNetV2, EfficientNetB3, ShuffleNet, SqueezeNet, and MnasNet—are known to be useful in many different situations. We prepared the dataset carefully by labeling and ensuring diversity, not just for animals, which makes these models handy for classifying images in various real-life scenarios. However, the field of machine learning keeps evolving, bringing newer and better models. To avoid our approach becoming outdated, we focus on established algorithms deliberately. By doing this, we lay a strong foundation that can set a standard for future experiments and advancements, ensuring that our work remains relevant and useful in the long term.

6.2.3 Conclusion Validity

Ensuring our study's findings are reliable means selecting appropriate methods to compare the results obtained. We used various measures like accuracy, precision, recall, and F1-score to assess model performance comprehensively. By employing these robust evaluation metrics and applying them correctly, we reached valid conclusions for our study. The combined metric approach, which integrates accuracy, inference time, and model size, further reinforces the reliability of our conclusions by providing a holistic assessment of model performance.

Chapter 7

Conclusions and Future Work

7.1 Conclusion

This research aimed to identify the most effective lightweight transfer learning models for animal image classification in resource-constrained environments. The literature review provided a foundational understanding of models such as MobileNetV2, EfficientNetB3, ShuffleNet, SqueezeNet, and MnasNet, focusing on their computational efficiency and accuracy. The experimental results showed effective learning and generalization for most models, though some exhibited signs of overfitting. A Friedman test confirmed significant differences in model performance. A combined metric approach, integrating accuracy, inference time, and model size, ranked ShuffleNet highest (0.854), followed by MnasNet (0.742), MobileNetV2 (0.571), SqueezeNet (0.410), and EfficientNetB3 (0.333).

The practical implications of the evaluated models are significant for applications in wildlife monitoring and conservation efforts. ShuffleNet, with the highest combined metric score, is ideal for mobile and embedded systems due to its balance of accuracy, low inference time, and small model size. MnasNet also shows strong potential for similar applications, offering robust performance across all metrics. MobileNetV2, while slightly lower in performance, remains a viable option with its respectable accuracy and moderate resource requirements. SqueezeNet and EfficientNetB3, despite scoring lower, provide valuable insights for specific use cases, particularly where minimizing model size or achieving high accuracy is critical.

In conclusion, this thesis presents a comprehensive methodology for selecting and evaluating lightweight transfer learning models for animal image classification in resource-constrained environments. The combined metric approach identified ShuffleNet and MnasNet as the most suitable models, balancing accuracy, efficiency, and model size. These findings provide valuable insights for future research and practical deployment strategies, ensuring that selected models are both accurate and resource-efficient. This research addresses a notable gap by providing practical guidance on selecting effective lightweight transfer learning models for animal image classification in resource-constrained scenarios. The reliability, generalizability, and adaptability of these models position them as significant contributors to advancements in wildlife monitoring and conservation efforts, bridging the gap between theoretical research and real-world application.

7.2 Future Work

This research contributes towards developing efficient and accurate models for animal image classification. However, there are opportunities to further improve the models and extend the scope of this work.

Future studies could focus on developing ensemble or hybrid models by strategically combining the architectures of different models investigated in this research. This approach holds promise to boost classification accuracy beyond what individual models can achieve. Further optimization of model performance can be pursued by experimenting with various preprocessing, data augmentation, and fine-tuning techniques. Techniques such as hyperparameter tuning, custom preprocessing pipelines, mixups, and cutouts could potentially enhance the robustness and accuracy of the models.

A practical next step involves integrating the best-performing models, such as ShuffleNet and MnasNet, into wildlife monitoring mobile and cloud applications. This will allow for the evaluation of real-time classification capabilities under resource constraints in field conditions. Additionally, collecting and working with larger, more diverse animal datasets can help evaluate how well the models generalize across different domains and species. This is crucial for real-world deployments involving new geographic regions and diverse species.

Addressing these directions would advance the models closer to practical deployments while continuing to push the boundaries of lightweight transfer learning for animal image classification.

References

[1] M. Ramasamy, M. A. Hannan, Y. A. Ahmed, and A. K. Dev, "Ann-based decision making in station keeping for geotechnical drilling vessel," vol. 9, p. 596, MDPI, 2021.

[2] L. Alzubaidi, J. Zhang, A. J. Humaidi, A. Al-Dujaili, Y. Duan, O. Al-Shamma, J. Santamaría, M. A. Fadhel, M. Al-Amidie, and L. Farhan, "Review of deep learning: Concepts, cnn architectures, challenges, applications, future directions," vol. 8, pp. 1–74, Springer, 2021.

[3] S. Srivastava, *Face Recognition Using Transfer Learning with VGG16*. Medium, 2020.

[4] M. Akay, Y. Du, C. L. Sershen, M. Wu, T. Y. Chen, S. Assassi, C. Mohan, and Y. M. Akay, "Deep learning classification of systemic sclerosis skin using the mobilenetv2 model," *IEEE Open Journal of Engineering in Medicine and Biology*, vol. 2, pp. 104–110, 2021.

[5] A. Soleimanipour, M. Azadbakht, and A. Rezaei Asl, "Cultivar identification of pistachio nuts in bulk mode through efficientnet deep learning model," vol. 16, pp. 2545–2555, Springer, 2022.

[6] X. Zhang, X. Zhou, M. Lin, and J. Sun, "Shufflenet: An extremely efficient convolutional neural network for mobile devices," in *Proceedings of the IEEE conference on computer vision and pattern recognition*, pp. 6848–6856, 2018.

[7] F. N. Iandola, S. Han, M. W. Moskewicz, K. Ashraf, W. J. Dally, and K. Keutzer, "Squeezenet: Alexnet-level accuracy with 50x fewer parameters and$<$ 0.5 mb model size," *arXiv preprint arXiv:1602.07360*, 2016.

[8] M. Tan, B. Chen, R. Pang, V. Vasudevan, M. Sandler, A. Howard, and Q. V. Le, "Mnasnet: Platform-aware neural architecture search for mobile," in *Proceedings of the IEEE/CVF conference on computer vision and pattern recognition*, pp. 2820–2828, 2019.

[9] B. Mahesh, "Machine learning algorithms-a review," *International Journal of Science and Research (IJSR).[Internet]*, vol. 9, no. 1, pp. 381–386, 2020.

[10] M. Shaha and M. Pawar, "Transfer learning for image classification," in *2018 second international conference on electronics, communication and aerospace technology (ICECA)*, pp. 656–660, IEEE, 2018.

[11] N. Houlsby, A. Giurgiu, S. Jastrzebski, B. Morrone, Q. De Laroussilhe, A. Gesmundo, M. Attariyan, and S. Gelly, "Parameter-efficient transfer learning for

nlp," in *International Conference on Machine Learning*, pp. 2790–2799, PMLR, 2019.

[12] F. Mehmood, S. Ahmad, and T. K. Whangbo, "Object detection based on deep learning techniques in resource-constrained environment for healthcare industry," in *2022 International Conference on Electronics, Information, and Communication (ICEIC)*, pp. 1–5, IEEE, 2022.

[13] B. Koonce, "Efficientnet," *Convolutional Neural Networks with Swift for Tensorflow: Image Recognition and Dataset Categorization*, pp. 109–123, 2021.

[14] M. Sandler, A. Howard, M. Zhu, A. Zhmoginov, and L.-C. Chen, "Mobilenetv2: Inverted residuals and linear bottlenecks." arXiv, 2018.

[15] L. Chen, S. Li, Q. Bai, J. Yang, S. Jiang, and Y. Miao, "Review of image classification algorithms based on convolutional neural networks," vol. 13, p. 4712, MDPI, 2021.

[16] E. K. Gyasi and S. Purushotham, "Advancements in soil classification: An in-depth analysis of current deep learning techniques and emerging trends," vol. 16, p. 11786221231214069, SAGE Publications Sage UK: London, England, 2023.

[17] V. Kamath and A. Renuka, "Deep learning based object detection for resource constrained devices-systematic review, future trends and challenges ahead," Elsevier, 2023.

[18] P. Flach, *Machine learning: the art and science of algorithms that make sense of data*. Cambridge university press, 2012.

[19] G. Englebienne, "Machine learning pattern recognition." Citeseer, 2016.

[20] G. James, D. Witten, T. Hastie, R. Tibshirani, and J. Taylor, "Unsupervised learning," in *An Introduction to Statistical Learning: with Applications in Python*, pp. 503–556, Springer, 2023.

[21] X. J. Zhu, "Semi-supervised learning literature survey." University of Wisconsin-Madison Department of Computer Sciences, 2005.

[22] L. P. Kaelbling, M. L. Littman, and A. W. Moore, "Reinforcement learning: A survey," *Journal of artificial intelligence research*, vol. 4, pp. 237–285, 1996.

[23] T.-T. Wong, "Performance evaluation of classification algorithms by k-fold and leave-one-out cross validation," vol. 48, pp. 2839–2846, Elsevier, 2015.

[24] B. J. Erickson and F. Kitamura, "Magician's corner: 9. performance metrics for machine learning models," vol. 3, p. e200126, Radiological Society of North America, 2021.

[25] M. Hu and F. You, "Research on animal image classification based on transfer learning," in *Proceedings of the 2020 4th International Conference on Electronic Information Technology and Computer Engineering*, pp. 756–761, 2020.

[26] D. F. C. Huertas, G. S. G. Gómez, and A. M. Á. Meza, "Image-based animal recognition based on transfer learning," *Scientia et Technica*, vol. 26, no. 03, pp. 406–411, 2021.

[27] K. Sanghvi, A. Aralkar, S. Sanghvi, and I. Saha, "Fauna image classification using convolutional neural network," *International journal of future generation communication and networking*, vol. 13, no. 1s, pp. 08–16, 2020.

[28] S. Taheri and Ö. Toygar, "Animal classification using facial images with score-level fusion," vol. 12, pp. 679–685, Wiley Online Library, 2018.

[29] Y. S. Kumar, N. Manohar, and H. Chethan, "Animal classification system: a block based approach," vol. 45, pp. 336–343, Elsevier, 2015.

[30] H. Yuan, S. Zhang, E. Qin, and H. Zhou, "Aquatic animal image classification technology based on transfer learning and data augmentation," *Journal of Coastal Research*, vol. 105, no. SI, pp. 129–133, 2020.

[31] F. Alharbi, A. Alharbi, and E. Kamioka, "Animal species classification using machine learning techniques," in *MATEC web of conferences*, vol. 277, EDP Sciences, 2019.

[32] H. Nguyen, S. J. Maclagan, T. D. Nguyen, T. Nguyen, P. Flemons, K. Andrews, E. G. Ritchie, and D. Phung, "Animal recognition and identification with deep convolutional neural networks for automated wildlife monitoring," in *2017 IEEE international conference on data science and advanced Analytics (DSAA)*, pp. 40–49, IEEE, 2017.

[33] M. Willi, R. T. Pitman, A. W. Cardoso, C. Locke, A. Swanson, A. Boyer, M. Veldthuis, and L. Fortson, "Identifying animal species in camera trap images using deep learning and citizen science," vol. 10, pp. 80–91, Wiley Online Library, 2019.

[34] Q. Zhang, X. Yi, J. Guo, Y. Tang, T. Feng, and R. Liu, "A few-shot rare wildlife image classification method based on style migration data augmentation," vol. 77, p. 102237, Elsevier, 2023.

[35] I. Zualkernan, S. Dhou, J. Judas, A. R. Sajun, B. R. Gomez, and L. A. Hussain, "An iot system using deep learning to classify camera trap images on the edge," vol. 11, p. 13, MDPI, 2022.

[36] B. Jiang, W. Huang, W. Tu, and C. Yang, "An animal classification based on light convolutional network neural network," in *2019 International Conference on Intelligent Computing and its Emerging Applications (ICEA)*, pp. 45–50, IEEE, 2019.

[37] B. Jiang, W. Huang, Y. Huang, C. Yang, and F. Xu, "Mpnet: A multiprocess convolutional neural network for animal classification," in *CCF Conference on Computer Supported Cooperative Work and Social Computing*, pp. 583–593, Springer, 2020.

[38] X. Liu, Z. Jia, X. Hou, M. Fu, L. Ma, and Q. Sun, "Real-time marine animal images classification by embedded system based on mobilenet and transfer learning," in *OCEANS 2019-Marseille*, pp. 1–5, IEEE, 2019.

[39] H. K. Kondaveeti, S. V. N. S. V. Guturu, K. S. Jayan Praveen, and S. V. S. Kumar, "A transfer learning approach to bird species recognition using mo-

bilenetv2," in *2023 7th International Conference on Intelligent Computing and Control Systems (ICICCS)*, pp. 787–794, 2023.

[40] T. B. Shahi, C. Sitaula, A. Neupane, and W. Guo, "Fruit classification using attention-based mobilenetv2 for industrial applications," vol. 17, p. e0264586, Public Library of Science San Francisco, CA USA, 2022.

[41] K. Sakthivelu, "Anidetnet using efficientnet-b3 model with transfer learning," vol. 12, pp. 4546–4553, 06 2023.

[42] H. Liu, "Animal image classification recognition based on transfer learning," vol. 7, pp. 135–140, Boya Century Publishing Limited, 2021.

[43] V. A. O. Nancy, P. Prabhavathy, M. S. Arya, and B. S. Ahamed, "Comparative study and analysis on skin cancer detection using machine learning and deep learning algorithms," *Multimedia Tools and Applications*, pp. 1–45, 2023.

[44] J. S. Byun, J. H. Lee, J. S. Kang, and B. S. Han, "Comparative analysis of imaging diagnostic models for tubular basophilia and mineralization of kidney," *Laboratory Animal Research*, vol. 38, no. 1, pp. 1–7, 2022.

[45] Y. Gulzar, Z. Ünal, H. Aktaş, and M. S. Mir, "Harnessing the power of transfer learning in sunflower disease detection: A comparative study," *Agriculture*, vol. 13, no. 8, p. 1479, 2023.

[46] P. Reshma, S. Saitejesh, V. Shashank, R. Jhansi, and V. A. Kumar, "Ocular disease identification using deep learning techniques," 2023.

[47] R. Chatterjee, A. Chatterjee, and R. Halder, "An efficient pneumonia detection from the chest x-ray images," in *Proceedings of International Conference on Machine Intelligence and Data Science Applications: MIDAS 2020*, pp. 779–789, Springer, 2021.

[48] A. Chalifah, S. Purbojo, R. Umitaibatin, and D. Rudiyanto, "Comparative study of convolutional neural network feature extractors used for covid-19 detection from chest x-ray images. 2020."

[49] G. Li, G. Erickson, and Y. Xiong, "Individual beef cattle identification using muzzle images and deep learning techniques," *Animals*, vol. 12, p. 1453, 06 2022.

[50] T. Feng, Y. Guo, X. Huang, J. Wu, and C. Cheng, "A method of body condition scoring for dairy cows based on lightweight convolution neural network," American Society of Agricultural and Biological Engineers, 2024.

[51] D. Avola, A. Bacciu, L. Cinque, A. Fagioli, M. R. Marini, and R. Taiello, "Study on transfer learning capabilities for pneumonia classification in chest-x-rays images," *Computer Methods and Programs in Biomedicine*, vol. 221, p. 106833, 2022.

[52] V. Shwetha, K. Prasad, C. Mukhopadhyay, B. Banerjee, and A. Chakrabarti, "Automatic detection of bacilli bacteria from ziehl-neelsen sputum smear images," in *2021 2nd International Conference on Communication, Computing and Industry 4.0 (C2I4)*, pp. 1–5, 2021.

[53] K. Saddami, N. Aulia, V. Maulidia, and Andriansyah, "Comparative analysis of lightweight pre-trained cnn models for coffee bean roasting level identification," in *2023 2nd International Conference on Computer System, Information Technology, and Electrical Engineering (COSITE)*, pp. 13–18, 2023.

[54] B. Syamsuri and I. Negara, "Plant disease classification using lite pretrained deep convolutional neural network on android mobile device," 12 2019.

[55] M. E. Karar, O. Reyad, M. Abd-Elnaby, A.-H. Abdel-Aty, and M. A. Shouman, "Lightweight transfer learning models for ultrasound-guided classification of covid-19 patients," *Comput. Mater. Contin*, vol. 69, pp. 2295–2312, 2021.

[56] A. K. Hrithik and V. Kumar, "Classification of fruit plants leaf and comparative analysis of machine learning and deep learning algorithms," in *2022 International Conference on Computing, Communication, and Intelligent Systems (ICCCIS)*, pp. 673–680, 2022.

[57] R. Sapkota, P. Sharma, and I. Mann, "Comparison of deep learning models for the classification of noctilucent cloud images," *Remote Sensing*, vol. 14, no. 10, p. 2306, 2022.

[58] B. E. Granger and F. Pérez, "Jupyter: Thinking and storytelling with code and data," *Computing in Science Engineering*, vol. 23, no. 2, pp. 7–14, 2021.

[59] S. Raschka and V. Mirjalili, *Python machine learning: Machine learning and deep learning with Python, scikit-learn, and TensorFlow 2.* Packt Publishing Ltd, 2019.

[60] S. Van Der Walt, S. C. Colbert, and G. Varoquaux, "The numpy array: a structure for efficient numerical computation," *Computing in science & engineering*, vol. 13, no. 2, pp. 22–30, 2011.

[61] W. McKinney *et al.*, "pandas: a foundational python library for data analysis and statistics," *Python for high performance and scientific computing*, vol. 14, no. 9, pp. 1–9, 2011.

[62] F. Chollet, *Deep learning with Python.* Simon and Schuster, 2021.

[63] A. Paszke, S. Gross, F. Massa, A. Lerer, J. Bradbury, G. Chanan, T. Killeen, Z. Lin, N. Gimelshein, L. Antiga, *et al.*, "Pytorch: An imperative style, high-performance deep learning library," *Advances in neural information processing systems*, vol. 32, 2019.

[64] S. Marcel and Y. Rodriguez, "Torchvision the machine-vision package of torch," in *Proceedings of the 18th ACM international conference on Multimedia*, pp. 1485–1488, 2010.

[65] N. Ari and M. Ustazhanov, "Matplotlib in python," in *2014 11th International Conference on Electronics, Computer and Computation (ICECCO)*, pp. 1–6, IEEE, 2014.

[66] S. Banerjee, "Animal image dataset - 90 different animals." Kaggle. Available online: `https://www.kaggle.com/datasets/iamsouravbanerjee/animal-image-dataset-90-different-animals/data`.

[67] “Google images.” `https://images.google.com/`.

[68] D. P. Kingma and J. Ba, “Adam: A method for stochastic optimization,” *arXiv preprint arXiv:1412.6980*, 2014.

[69] S. Santurkar, D. Tsipras, A. Ilyas, and A. Madry, “How does batch normalization help optimization?,” *Advances in neural information processing systems*, vol. 31, 2018.

[70] H. Ghorbani, “90 different animals image classification 92.5%.” Kaggle Code Repository, 2023. Available online: `https://www.kaggle.com/code/hamedghorbani/90-different-animals-image-classification-92-5#Training-:-Fine-Tuning`.

[71] C. López-Vázquez and E. Hochsztain, “Extended and updated tables for the friedman rank test,” *Communications in Statistics-Theory and Methods*, vol. 48, no. 2, pp. 268–281, 2019.

Faculty of Faculty, Blekinge Institute of Technology, 371 79 Karlskrona, Sweden