ADVANCED INTELLIGENT SYSTEMS
Open Access
www.advintellsyst.com

Green AI-Driven Concept for the Development of Cost-Effective and Energy-Efficient Deep Learning Method: Application in the Detection of *Eimeria* Parasites as a Case Study

Suheda Semih Acmali, Yasin Ortakci,* and Huseyin Seker*

Although large-scale pretrained convolutinal neural networks (CNN) models have shown impressive transfer learning capabilities, they come with drawbacks such as high energy consumption and computational cost due to their potential redundant parameters. This study presents an innovative weight-level pruning technique that mitigates the challenges of overparameterization, and subsequently minimizes the electricity usage of such large deep learning models. The method focuses on removing redundant parameters while upholding model accuracy. This methodology is applied to classify *Eimeria* species parasites from fowls and rabbits. By leveraging a set of 27 pretrained CNN models with a number of parameters between 3.0M and 118.5M, the framework has identified a 4.8M-parameter model with the highest accuracy for both animals. The model is then subjected to a systematic pruning process, resulting in an 8% reduction in parameters and a 421M reduction in floating point operations while maintaining the same classification accuracy for both fowls and rabbits. Furthermore, unlike the existing literature where two separate models are created for rabbits and fowls, this article presents a combined model with 17 classes. This approach has resulted in a CNN model with nearly 50% reduced parameter size while retaining the same accuracy of over 90%.

1. Introduction

Deep learning models have been used in several domains due to their generalizable prediction ability.[1,2] Convolutional neural networks (CNNs) are the cornerstone models in deep learning, producing state-of-the-art results in many fields, especially in image processing. These models have shown remarkable efficiency, excelling in object detection and image classification tasks by recognizing intricate patterns in different regions of images.[1] In the last decade, large-scale, general-purpose pretrained CNN models with millions of parameters trained with extensive image datasets have emerged. By exploiting the diversity of these datasets and the abundance of parameters, these models provide a general-purpose transfer learning capability in image processing. Users can achieve significantly higher accuracy by fine-tuning these models for target domains, rather than training custom CNN models from scratch.

Although pretrained CNN models exhibit impressive performance, they have some drawbacks, such as substantial energy consumption, high computational costs, and long running times. The primary factors causing these challenges are the extensive number of parameters present in these models, particularly within the convolutional layers, coupled with the resource-intensive computations carried out on these parameters.[3] For instance, EfficientNetV2-L[4] has ≈120 million parameters and requires 53 billion floating point operations (FLOPs) to process a single image. As the parameters in these deep architectures are tailored for general-purpose usage, they are redundant when applied to a specific domain. This redundancy incurs unnecessary energy consumption and computational overhead, even though these parameters do not contribute to generating outcomes. Reducing the computational complexities is vital in these models to achieve energy

---

S. S. Acmali
Department of Computer Technologies
Zonguldak Bulent Ecevit University
İncivez, Üniversite Cd, 67100 Zonguldak, Turkey
E-mail: semih.acmali@beun.edu.tr

The ORCID identification number(s) for the author(s) of this article can be found under https://doi.org/10.1002/aisy.202300644.

DOI: 10.1002/aisy.202300644

Y. Ortakci
Computer Engineering Department
Karabük University
Balıklar Kayası Mevkii, Merkez, 78050 Karabük, Turkey
E-mail: yasinortakci@karabuk.edu.tr

H. Seker
The Faculty of Computing, Engineering and The Built Environment
Birmingham City University
15 Bartholomew Row, Birmingham B5 5JU, UK

efficiency.[5] Likewise, to enable the deployment of these models on resource-constrained devices, a reduction in the number of parameters is essential.[6,7]

To address the challenge of overparameterization in these models, researchers have explored various methods for compressing neural networks. Among these techniques, one notable approach is pruning, which involves the removal of weights having minimal or negligible influence on the model's performance. Pruning the weight parameters in pretrained CNN models, especially within the convolutional layer that includes a significant portion of parameters, is critical for both the success and energy efficiency of the models.[8] The pruning approaches should strike a balance between preserving the accuracy and reducing the number of parameters. In line with this objective, this study introduces a weight-level pruning method designed to remove ineffective parameters from the convolutional layer, all while maintaining the accuracy of pretrained CNN models. This method ranks the weights across all filters based on their contributions to the model's outputs. By selectively eliminating a certain proportion of weights with the least impact, this approach effectively mitigates the issues of overparameterization and high energy consumption in these expansive models.

On the other hand, this study specifically focused on classifying *Eimeria* parasites, leading to serious diseases in many animals, especially rabbits and fowls.[9] Our approach involved leveraging pretrained CNN models to identify these parasites, which have many species. In this context, we created a framework employing 27 different pretrained models widely used in contemporary research and industry. In the initial phase, this framework determined four models that produced the most successful results among the options. Subsequently, from this subset of four, it then found the most effective model that could be used to detect *Eimeria* parasites with further analysis. Throughout this process, we exclusively used these models for inference purposes and did not perform any fine-tuning in the convolutional layer. Our efforts were solely concentrated on training the classification layers of these versatile models, customizing them to suit the nuances of *Eimeria* species. In the next phase, we reduced the model size by implementing our pruning technique into the highest success model. Our method gradually removes the irrelevant parameters by evaluating the model's accuracy. This systematic approach effectively discarded redundant parameters while preserving the model's accuracy.

The results show that reducing the number of parameters up to 8% does not have a negative effect on *Eimeria* parasite classification success. As in many other studies on this subject, the employment of fine-tuning after pruning would further enhance this rate. However, our study differs from others in its distinct aim to achieve the same results by both reducing the number of parameters and minimizing energy consumption without fine-tuning. This choice is due to the fact that retraining the convolutional layers of these models for fine-tuning can take days and be very energy-consuming. The proposed framework can perform *Eimeria* classification with minimal energy consumption without compromising accuracy. In addition, in contrast to models developed separately for fowls or rabbits in the literature, we proposed a model to identify all *Eimeria* species causing illness in both fowls and rabbits. Thus, we have presented a pragmatic and energy-conscious solution for practitioners to apply directly in the field. The key contributions of our study are as follows: 1) This study presents a comprehensive analysis comparing the transfer learning performance of 27 different contemporary CNN pretrained models for *Eimeria* parasite classification and identifies the most proficient one. 2) As a novel approach, this study proposes a weight-level pruning methodology capable of reducing model weights without the need for fine-tuning and provides energy efficiency in the model's usage. 3) This innovative technique holds universality, being applied to all pretrained CNN models regardless of the architecture. 4) This study also extends its impact by introducing a universal *Eimeria* species classification model. Unlike existing studies that can only classify *Eimeria* species in fowls or rabbits, this holistic approach effectively recognizes all *Eimeria* parasites across the spectrum.

2. Related Work

Due to the dual focus of our study encompassing network compression and the classification of *Eimeria* parasites, the related work section is subdivided into two segments. Each segment is dedicated to addressing pertinent research on these respective topics.

2.1. Compression of Deep Learning Models

In recent years, researchers have delved into various strategies for tackling overparameterization and redundancy issues in pretrained CNN models. These strategies, which aim to downsize models, center around network compression techniques, broadly classified into knowledge distillation, parameter quantization, and network pruning.[10]

Knowledge distillation involves transferring the problem-solving capabilities of a complex and large "teacher" model to a more straightforward and smaller size "student" model. This approach gained prominence through Hinton et al.'s successful distillation of extensive ensemble models into a compact model with improved accuracy,[11] inspiring subsequent research. For instance, some attempts leveraging this approach for cancer cell detection with pretrained CNN models[12,13] have yielded close prediction accuracy while significantly reducing the dimensions of the teacher model. Moreover, Luo et al. demonstrated even surpassing the accuracy of the primary model in face recognition using knowledge distillation.[14]

Parameter quantization involves modifying the weights to be represented by fewer bits to reduce model size. Using this approach, Han et al. achieved a 49-fold reduction in the size of VGG-16,[15] while Gong et al. compressed their CNN models by 32 times with only a 1% loss in accuracy.[16] However, this technique does not reduce the number of FLOPs, thereby becomes ineffective in accelerating models.

Network pruning, which aims to reduce the computational cost and energy expenditure of models with acceptable performance degradation, eliminates ineffective or minimally contributing parameters within a network. Pruning endeavors have primarily focused on two application domains: weight or filter removal. Weight-level pruning often entails a threshold-based approach for eliminating weights below a specific value.[17,18] While most of the attention has focused on the dilution of weights in the convolutional layer, refs. [19,20] have targeted the reduction of weight in the fully connected layer. On the other hand, filter-level pruning research consists of the studies that remove least-contributing filters[21–26] or the filters which have similar content.[27–29] Among the existing compression methods, network pruning is the most preferred method due to its simplicity and high compression capacity.[30]

All these methods involve fine-tuning the models after compression to ensure that the generated results are similar to those of the models' original versions. Nevertheless, these approaches carry a significant energy overhead during the fine-tuning phase, often spanning several days or more. Differing from these methodologies, our focus is to attain results on par with the original model, eliminating the necessity for fine-tuning after pruning. Our primary objective is to achieve this outcome with minimal energy consumption.

2.2. Classification of *Eimeria* Parasites

Eimeria is a parasitic organism that invades the intestines of domestic animals. If it is not detected at an early stage, it spreads rapidly and leads to the mortality of these animals. This parasite has numerous species, each causing distinct pathogenic and virulence effects on the host animals. Therefore, identifying these specific species holds significant importance in disease treatment. Even though *Eimeria* species show morphological differences in terms of color, size, contour, wall thickness, and internal structure, the identification of these species by the human eye within microscopic images is highly error-prone.[31] This issue creates a demand for automated systems that can categorize *Eimeria* parasites with a wide range of species. Despite the studies that employ DNA sequences for species detection in the literature,[32] researchers have recently turned to image analysis methods that provide results with notably reduced cost and quicker turnaround.

In this context, Castañón et al. achieved 85.75% prediction accuracy with a Bayesian classifier utilizing a Gaussian distribution model in the real-time diagnosis of fowl *Eimeria* species.[31] In the case of rabbit species, Abdalla et al. proposed a pixel-based feature extraction method to categorize the species in this group through the K-nearest neighbor (KNN).[33] Abdalla and Seker expanded their investigation by using an artificial neural network to identify *Eimeria* species, yielding more favorable outcomes than KNN.[34] Similarly, Buyukyilmaz et al. used a multilayer perceptron deep learning model to detect chicken *Eimeria* species.[35] While refs. [36,37] incorporated CNN models to classify the fowl species, Boufenar et al. compared five different pretrained CNN models for the same purpose and found the Xception as the most proficient model.[9]

Our study adopts a holistic approach to identify all varieties of *Eimeria* species leading to disease in both fowls and rabbits, diverging from the conventional practice of detecting them within distinct groups. Furthermore, this study not only determines the most effective pretrained model out of 27 diverse models within this domain but also makes it energy-efficient by implementing our pruning techniques.

3. Experimental Section

This section thoroughly explains our methodology, centered around a four-stage framework (**Figure 1**). In the initial phase, we preprocessed the *Eimeria* parasite image dataset associated with rabbit and fowl diseases, utilizing various image preprocessing techniques. Subsequently, we engaged in model selection to identify the optimal performing model within a pool of 27 distinct, contemporary, pretrained CNN models. Our focus then shifted to the development of an energy-efficient model through weight pruning on the previously determined best-performing model. In the final step, we conducted an evaluation of both the pruned and original versions of this model, utilizing a comprehensive dataset covering species from both rabbits and fowls.

3.1. Data Preprocessing

The dataset used in this study comprises seven species affecting fowls and ten species affecting rabbits of *Eimeria* parasites. **Figure 2** depicts microscopic images of these 17 species, with heights ranging from 192 to 642 pixels and widths from 177 to 492 pixels. These rectangular images contain oval-shaped cells and a background. We applied a sequence of segmentation procedures to isolate cells from the background. To ensure standardization, we placed all cells in the center of a black canvas whose dimensions are 528 pixels in width and 642 pixels in height. Additionally, inconsistent lighting during microscopic imaging can significantly impact results.[3] Therefore, to minimize

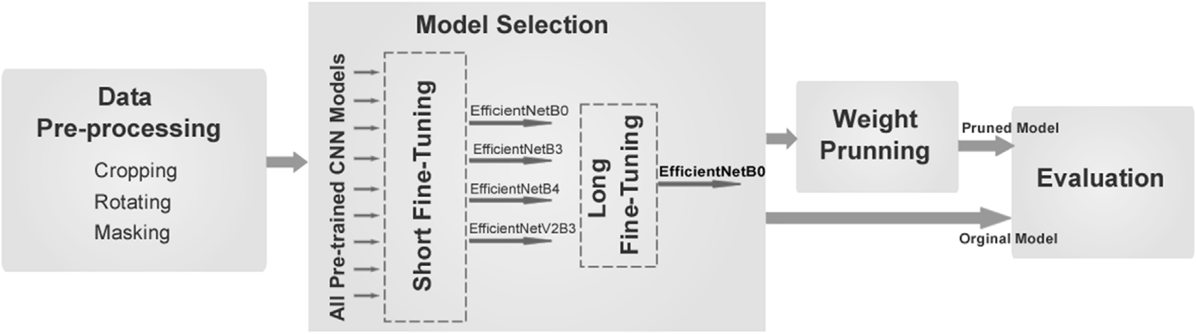

Figure 1. Overview of our methodology—phase 1: data preprocessing, phase 2: model selection (including short fine-tuning and long fine-tuning), phase 3: weight-level pruning of CNN models, and phase 4: comparison of pruned and original models.

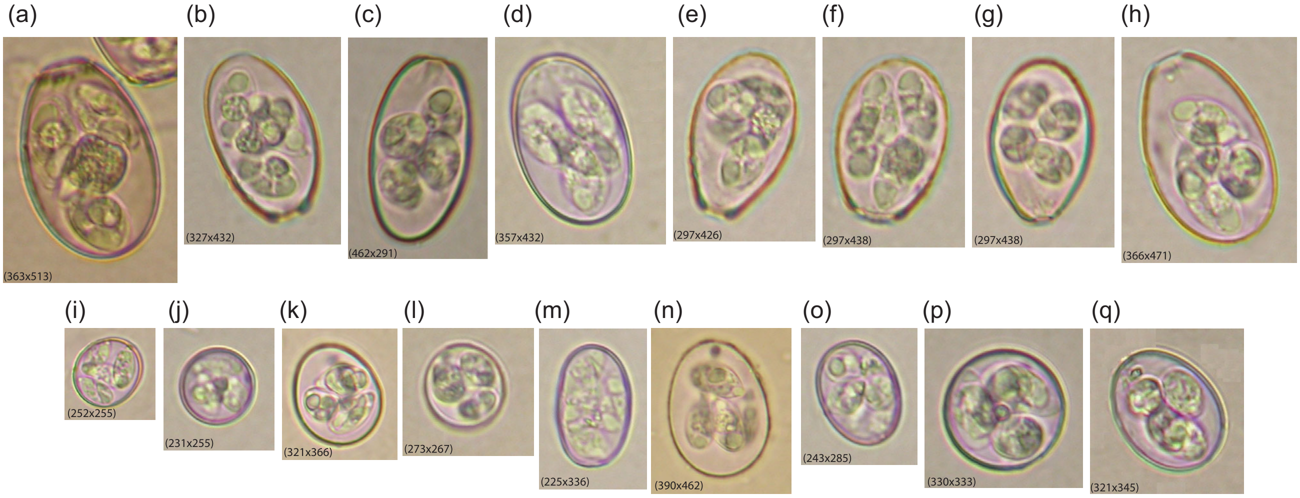

Figure 2. *Eimeria* species causing disease in fowls and rabbits; a) Magna, b) Coecicola, c) Vejdovskyi, d) Flavescens, e) Intestinalis, f) Media, g) Piriformis, h) Stiedai, i) Necatrix, j) Exigua, k) Brunetti, l) Mitis, m) Perforans, n) Maxima, o) Acervulina, p) Praecox, and q) Tenella. The size of all images is given in (*x*,*y*) just below each image, *x*: width and *y*: height.

illumination variations that can distort results, we converted all images to grayscale.

We applied the Otsu image binarization algorithm to pinpoint cell edges within the images.[38] This method identifies contours representing cell boundaries and picks the contour with the largest area as the representative cell. For data preprocessing purposes, we applied the following operations on the original image, as illustrated in **Figure 3**. 1) Cropping: It extracts the cell within the image by utilizing the previously determined contour. 2) Rotating: It aligns the cropped cell into an upright orientation. 3) Masking: This procedure utilized the dimensions of the largest cell within the dataset as a benchmark. These dimensions are defined as the reference size of the canvas for all other images. Each image was then positioned within a canvas of this size, ensuring uniformity throughout the dataset. Any areas outside of the cell boundary were masked with black color, thereby creating a consistent visual representation across images.

3.2. Selection of Pretrained CNN Model

As shown in **Figure 4**, CNN comprises three main components: 1) an input layer to receive data, 2) a convolutional network responsible for feature extraction, and 3) a classifier head layer to perform the classification task.[39] The fine-tuning of the convolutional network is the most energy-demanding and computation-intensive stage of CNN models, which has a substantial impact on overall model performance.[3] To avoid this overhead, we utilize the large pretrained CNN models that have been extensively trained on vast image datasets for general use and could produce high-accuracy results. Thereby, we saved time and energy to be spent on training the convolutional network.

The model selection involves two distinct stages aimed at identifying the most effective pretrained CNN model. The first stage entails a short fine-tuning step spanning of five epochs only. During this phase, we modified only the weight parameters within the classification layer and kept the rest of the model parameters the same. This fine-tuning procedure is applied across all models listed in Table 2. The models with the highest classification accuracy went on to the next step, long fine-tuning, in which we ran an extended fine-tuning spanning 200 epochs to drill their performance. In contrast to previous studies, our approach involves freezing the weight parameters in the convolutional layers without interfering with them during both short and long fine-tuning sessions. Thus, this strategy ensures that these models efficiently detect *Eimeria* species with minimal energy

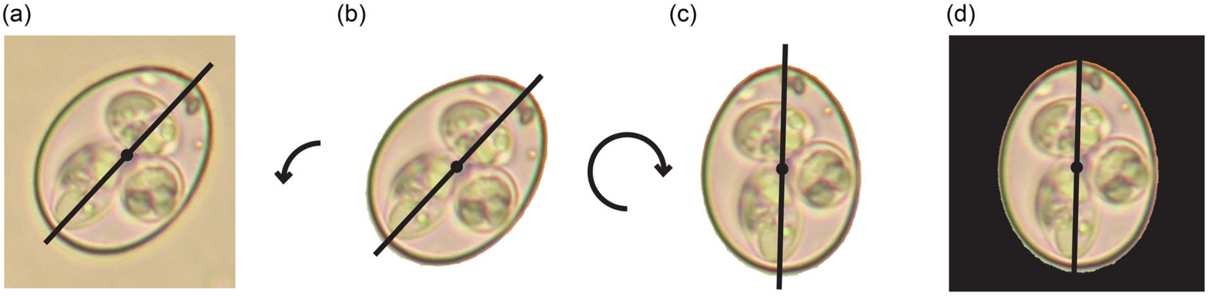

Figure 3. Image preprocessing steps: a) cropping, b) and c) rotating, and d) masking.

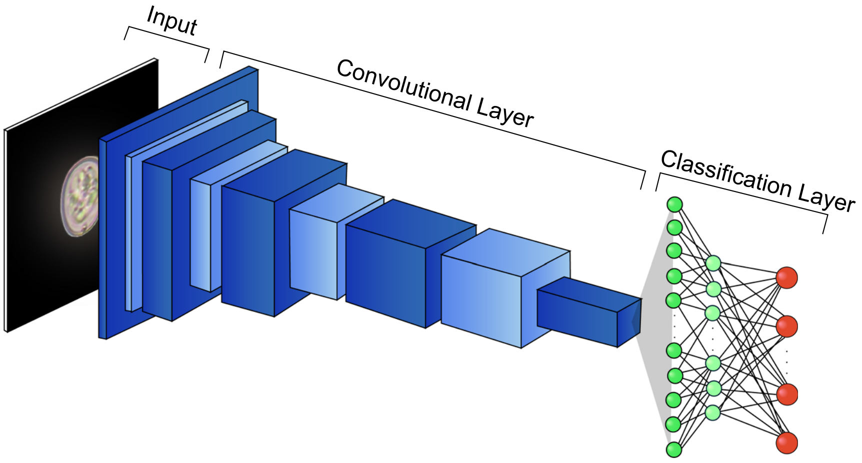

Figure 4. The general structure of CNN model including the input layer, convolutional layer, and fully connected classification layer.

consumption by not updating the convolutional layer which has high energy consumption.

3.3. Pruning

After determining the optimal model, this section describes our proposed weight-level pruning technique. This innovative method serves to further curtail computational complexity and diminish energy consumption within the model. Essentially, this technique targets the elimination of unproductive weights situated within the convolutional layer of the CNN.

Let us consider a convolutional network of a CNN comprising L layers. In this network, we parameterized the weights of a specific layer(l) with $W(l) \in \mathrm{R}^{M_l \times N_l \times K \times K}$, $1 \leq l \leq L$ representation (**Figure 5**). For layer l, M_l signifies the number of channels in the input feature map, while N_l represents the number of channels in the output feature map. $K \times K$ is the height and width of the kernel holding the trainable weights. Equation (1) illustrates $W(l)^{m,n}$, depicting a kernel matrix applied between channel m in the input map and channel n in the output map at layer l of the model, where $1 \leq m \leq M_l, 1 \leq n \leq N_l$.

$$W(l)^{m,n} = \begin{pmatrix} w_{1,1} & \cdots & w_{1,K} \\ \vdots & \ddots & \vdots \\ w_{K,1} & \cdots & w_{K,K} \end{pmatrix} \tag{1}$$

Our proposed technique focuses on a productivity analysis of all trainable weights in all convolutional layers, aiming to eliminate ineffective ones. To achieve this, we flatten the matrix $W(l)_{i,j}^{m,n}$, which contains all weights between input and output feature map channels in the corresponding layer, into a vector W' as in Equation (3).

$$W(l)_{i,j}^{m,n} = \begin{pmatrix} w_{1,1}^{1,1} & \cdots & w_{1,K}^{1,1} & \cdots & w_{1,1}^{1,M} & \cdots & w_{1,K}^{1,M} \\ \vdots & \ddots & \vdots & \cdots & \vdots & \ddots & \vdots \\ w_{K,1}^{1,1} & \cdots & w_{K,K}^{1,1} & \cdots & w_{K,1}^{1,M} & \cdots & w_{K,K}^{1,M} \\ \vdots & \ddots & \vdots & \cdots & \vdots & \ddots & \vdots \\ w_{1,1}^{N,1} & \cdots & w_{1,K}^{N,1} & \cdots & w_{1,1}^{N,M} & \cdots & w_{1,K}^{N,M} \\ \vdots & \ddots & \vdots & \cdots & \vdots & \ddots & \vdots \\ w_{K,1}^{N,1} & \cdots & w_{K,K}^{N,1} & \cdots & w_{K,1}^{N,M} & \cdots & w_{K,K}^{N,M} \end{pmatrix} \tag{2}$$

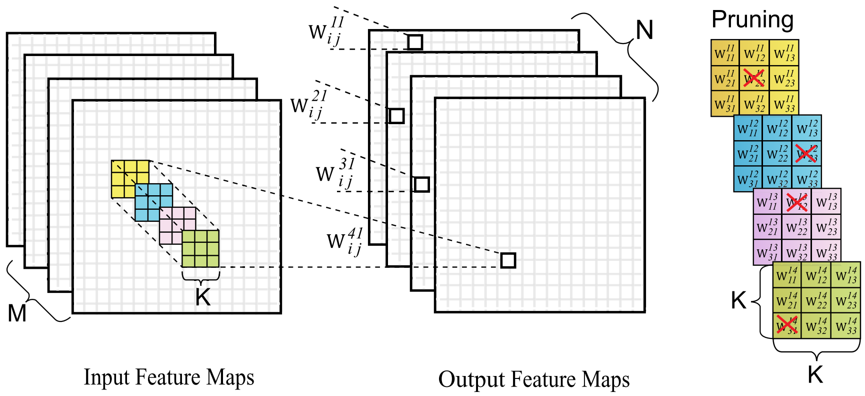

Figure 5. Pruning the weights in a convolutional layer (l) in the model.

$$W' = \left(w_{1,1}^{1,1}, \ldots, w_{K,K}^{1,1}, \ldots, w_{1,1}^{1,M}, \ldots, w_{K,K}^{1,M}, \ldots, w_{1,1}^{N,1}, \ldots, w_{K,K}^{N,1}, \ldots, w_{K,K}^{N,M}\right) \tag{3}$$

The vector W' is then sorted based on absolute values of weights using the argsoft function as indicated in Equation (4). Subsequently, we obtained the ordered list, $\overline{W}$, and defined the threshold (T) value to identify the productive weights (Equation (5)). The limit in Equation (5) is the index that determines T in the weight vector ($\overline{W}$) and the limit value is calculated as shown in Equation (6).

$$\overline{W} = \text{argsort}(|W'|) \tag{4}$$

$$T = \overline{W}[\text{limit}] \tag{5}$$

$$\text{limit} = \dim[W'] * p/100 \tag{6}$$

where $\dim[W'] = M_l \times N_l \times K \times K$ (total number of weights in the layer) and p is the percentage pruning rate, we applied.

Algorithm 1. The proposed weight pruning method.

```
Require: unpruned model, dataset, the accuracy of unpruned model ($Highest\_ACC$)
Ensure: pruned model with a $p$-value
for $p = 1$ to 100 do
    for each $l$ in $L(layers)$ do
        for each $n$ in $N(output feature map channels)$ do
            for each $m$ in $M(input feature map channels)$ do
                for $i = 1$ to $K(kernel height)$ do
                    for $j = 1$ to $K(kernel width)$ do
                        Add $W_{i,j}^{m,n}$ to $W'$
                    end for
                end for
            end for
        end for
        for each $n$ in $N$ do
            for each $m$ in $M$ do
                for $i = 1$ to $K$ do
                    for $j = 1$ to $K$do
                        if $W_{i,j}^{m,n} < T$ then
                            $W_{i,j}^{m,n} = 0$
                        end if
                    end for
                end for
            end for
        end for
    end for
    $ACC$=Calculate accuracy
    if $ACC = Highest\_ACC$ then
        Keep the model and $p$-value
    end if
end for
```

In order to determine the optimal value of p, we conducted an iterative experiment in which we ran the CNN model for all integer values of p within the range of $[1, 100]$. Based on the results of these experiments, we selected the highest p value that did not cause a decrease in the accuracy of the model as the limit index. Once the limit value was determined in parallel with the highest p value, we reset all weights whose index is less than limit in $\overline{W}$ to zero. These weights, which are below T value, were deemed to be ineffective and are removed as in Equation (7).

$$W_{i,j} = \begin{cases} 0 & if \quad W_{i,j}^{m,n} < T \\ W_{i,j}^{m,n} & \text{otherwise} \end{cases} \tag{7}$$

Figure 5 provides a visual representation of our proposed pruning approach, and Algorithm 5 outlines the detailed pseudocode describing its operational steps.

4. Experimental Results

This section consists of analyses to determine the optimal pretrained model for effectively classifying *Eimeria* parasites. Furthermore, it assesses the impact of our proposed pruning method on the optimum model's performance. Our study encompasses the utilization of three distinct datasets. The first dataset includes microscopic images of seven distinct *Eimeria* species causing diseases in domestic fowls.[31] The second one comprises images of ten diverse *Eimeria* species affecting rabbits.[40] **Table 1** lists the species in the datasets and their sample numbers. Additionally, a combined dataset containing both sets

Table 1. The number of *Eimeria* species samples in the datasets affecting fowls and rabbits.

Dataset	*Eimeria* species	Number of images
Rabbit	Coecicola (COE)	192
	Exigua (EXI)	290
	Flavescens (FLA)	379
	Intestinalis (INT)	115
	Magna (MAG)	420
	Media (MED)	190
	Perforans (PER)	151
	Piriformis (PIR)	125
	Stiedai (STI)	96
	Vejdovskyi (VEJ)	264
	Total	2222
Fowl	Acervulina (ACE)	726
	Brunetti (BRU)	435
	Maxima (MAX)	339
	Mitis (MIT)	795
	Necatrix (NEC)	468
	Praecox (PRA)	857
	Tenella (TEN)	605
	Total	4225

of images has been assembled, denoted as the "all-species" dataset in subsequent sections. This composite dataset serves as the foundation for classifying all *Eimeria* species, constituting a key focus of our investigation. Despite the imbalanced distribution of sample numbers among species in the datasets, our preliminary study revealed that this imbalance did not negatively impact the classification results. Consequently, similar to other studies[9,31,33,35–37] in the literature, we utilized the datasets without employing any sampling methods to balance them.

During the short fine-tuning, we employed a classification layer composed of two dense layers featuring 512 and 256 neurons, ending in 17 output units. We set the dropout rate to 0.3 and defined the fine-tuning parameters for the model as follows: $learning\ rate = 0.0001,\ batch\ size = 128,\ activation\ function = “Softmax,”\ and\ optimizer = “Adam”$. We conducted all the tests using a fivefold cross-validation, which splits the dataset into five equal parts and trains the model five times, in each iteration using four parts for training and one part for validation.

We used accuracy, recall, precision, and F1-score metrics to gauge the classification performance. To calculate these metrics, we utilized true positive (TP), true negative (TN), false positive (FP), and false negative (FN) values obtained from the confusion matrix. Accuracy (*Acc*) is the ratio of accurately predicted species to the total species (Equation (8)). Recall represents the proportion of species correctly identified as TP relative to the total actual positive species (Equation (9)). Precision stands for the portion of species correctly identified as TP among all species predicted as positive (Equation (10)). The F1-score is derived as the harmonic mean of recall and precision, balancing between these two values (Equation (11)). These metrics fall within the range of $[0, 1]$, where higher values indicate a higher classification performance.

$$\text{Accuracy} = (\text{TP} + \text{TN}) / (\text{TP} + \text{TN} + \text{FP} + \text{FN}) \tag{8}$$

$$\text{Recall} = \text{TP} / (\text{TP} + \text{FN}) \tag{9}$$

$$\text{Precision} = \text{TP} / (\text{TP} + \text{FP}) \tag{10}$$

$$\text{F1 Score} = 2 * (\text{Precision} * \text{Recall}) / (\text{Precision} + \text{Recall}) \tag{11}$$

4.1. Model Selection Results

In the initial phase of model selection, we administered the short fine-tuning procedure to 27 pretrained models on the all-species dataset. **Table 2** presents the minimum (Min), maximum (Max), mean, median, and standard deviation (SD) values of the models' classification accuracies. Among these models, the ones demonstrating a mean accuracy of 0.90 or higher, namely, EfficientNetB0 (0.91), EfficientNetV2B3 (0.90), EfficientNetB3 (0.90), and EfficientNetB4 (0.90), proceeded to the subsequent long fine-tuning phase. These four models were subjected to long fine-tuning with the same hyperparameters used in short fine-tuning, with outcomes showcased in **Table 3**. Ultimately, the most proficient model in *Eimeria* parasite classification emerged as EfficientNetB0, boasting a remarkable achievement mean accuracy rate of 0.966.

EfficientNetB0 serves as the foundational model in the EfficientNet series, spanning versions B0 through B7. It comprises an initial layer and seven blocks, with the number of layers progressively increasing from B0 to B7. For instance, B0 contains 237 layers, while B7 boasts 813 layers. EfficientNetB0 has significantly fewer parameters not only compared to other models in the EfficientNet series but also compared to other pretrained CNN models such as DenseNet, ResNet, Xception, Inception, and NasNET. This attribute contributes to its status as a highly efficient pretrained model, as it executes fewer FLOPs than its alternatives.

Table 2. Classification accuracies of all pretrained models in the short fine-tuning (five epochs). Bold indicate the most successfull model.

Pretrained model	Parameter [million]	Min	Max	Mean	Median	SD
DenseNet121[42]	7.6	0.68	0.78	0.74	0.76	0.033
DenseNet169[42]	13.6	0.68	0.77	0.74	0.74	0.027
DenseNet201[42]	19.4	0.65	0.82	0.76	0.77	0.046
EfficientNetB0[41]	4.8	0.88	0.93	**0.91**	0.90	0.016
EfficientNetB1[41]	7.3	0.86	0.91	0.88	0.88	0.015
EfficientNetB2[41]	8.6	0.84	0.91	0.87	0.87	0.022
EfficientNetB3[41]	11.7	0.85	0.91	**0.90**	0.90	0.016
EfficientNetB4[41]	18.7	0.87	0.93	**0.90**	0.90	0.015
EfficientNetB5[41]	29.6	0.87	0.91	0.89	0.89	0.012
EfficientNetB6[41]	42.2	0.79	0.86	0.82	0.82	0.022
EfficientNetB7[41]	65.5	0.83	0.91	0.89	0.89	0.022
EfficientNetV2B0[4]	6.7	0.83	0.90	0.87	0.86	0.025
EfficientNetV2B1[4]	7.7	0.81	0.89	0.87	0.87	0.024
EfficientNetV2B2[4]	9.6	0.86	0.92	0.88	0.88	0.020
EfficientNetV2B3[4]	13.8	0.86	0.91	**0.90**	0.90	0.017
EfficientNetV2S[4]	21.1	0.86	0.90	0.89	0.89	0.014
EfficientNetV2M[4]	53.9	0.77	0.85	0.82	0.82	0.020
EfficientNetV2L[4]	118.5	0.78	0.85	0.82	0.82	0.022
InceptionResNetV2[43]	55.2	0.13	0.18	0.16	0.16	0.016
InceptionV3[44]	22.9	0.60	0.71	0.66	0.66	0.035
MobileNet[45]	3.8	0.79	0.86	0.82	0.82	0.021
MobileNetV2[46]	3	0.71	0.82	0.78	0.78	0.034
NasNetLarge[47]	87.1	0.53	0.63	0.57	0.56	0.027
ResNet50V2[48]	24.7	0.39	0.65	0.50	0.50	0.084
ResNet101V2[48]	43.8	0.35	0.57	0.45	0.43	0.070
ResNet152V2[48]	59.5	0.26	0.49	0.35	0.34	0.066
Xception[49]	22	0.53	0.61	0.57	0.57	0.025

Table 3. Classification accuracies of the most successful four pretrained models in the long fine-tuning (200 epochs). Bold indicate the most successfull model.

Pretrained model	Weight [million]	Min	Max	Mean	Median	SD
EfficientNetB0	4.8	0.958	0.972	**0.964**	0.966	0.006
EfficientNetB3	11.7	0.936	0.952	0.946	0.946	0.007
EfficientNetB4	18.7	0.939	0.948	0.944	0.945	0.009
EfficientNetV2B3	13.8	0.958	0.966	0.962	0.961	0.004

The architecture of EfficientNetB0 is distinguished by several innovative features, including Inverted Bottleneck Residual Blocks (MBConv), Squeeze-and-Excitation (SE) blocks, and the Compound Scaling method.[41] MBConv introduces a significant paradigm shift in convolution implementation by breaking down standard convolutions into depthwise and pointwise convolutions, resulting in reduced computational cost and parameter count without compromising accuracy. SE blocks boost the model performance by selectively emphasizing informative features and suppressing less useful ones. Meanwhile, Compound Scaling balances the network by scaling the dimensions of depth, width, and resolution. These combined features position EfficientNetB0 as a compelling choice for fast computation, particularly in mobile and real-time environments.

4.2. Model Pruning Results

To assess the impact of pruning on EfficientNetB0, we evaluated its performance across three datasets. This assessment involved gradually increasing the pruning ratio from 1 to 100. **Table 4** presents the weight pruning ratios along with the resultant classification outcomes linked to these pruning rates. Our proposed method achieved the same accuracy as the unpruned model with a weight pruning of 5% on the rabbit dataset, eliminating ≈189 K parameters and 209 million (M) FLOPs. Similarly, the fowl dataset yielded the same accuracy as the original model with a 7% pruning rate, leading to the removal of roughly 261 K parameters and 370M FLOPs. Within the all-species dataset, the model maintained its accuracy without compromising up to an 8% pruning rate, discarding around 301 K parameters and 421M FLOPs. The number of FLOPs directly influences the computational complexity of a model. By reducing FLOPs, the number of operations in the model decreases, resulting in fewer multiply–accumulate operations performed on the CPU. Additionally, pruning the weight parameters leads to a smaller model size, reducing the memory footprint and the amount of data flowing between memory and the CPU. These two key optimizations contribute to a reduction in the energy consumption of the model, thereby enhancing its energy efficiency.

Table 4. Changes in classification results and reduction in the weight and FLOP numbers of the model in response to alterations in the pruning ratio.

Dataset	Pruning ratio [%]	Reduced weight number [K]	Reduced FLOP number [M]	Accuracy	Recall	Precision	F1-score
Rabbit	0	0	0	97%	0.97	0.97	0.97
	5	**189 K**	**209M**	**97%**	**0.96**	**0.95**	**0.96**
	10	377 K	403M	93%	0.92	0.92	0.90
	25	943 K	956M	79%	0.75	0.73	0.71
	50	1,887 K	1,914M	23%	0.23	0.08	0.11
	75	2,830 K	2,868M	14%	0.10	0.01	0.02
Fowl	0	0	0	95%	0.96	0.95	0.95
	7	**261 K**	**370M**	**95%**	**0.95**	**0.95**	**0.95**
	10	377 K	515M	92%	0.93	0.92	0.92
	25	943 K	1,222M	83%	0.84	0.87	0.83
	50	1,887 K	2,445M	31%	0.28	0.19	0.20
	75	2,830 K	3,664M	12%	0.14	0.02	0.03
All-species	0	0	0	96%	0.96	0.96	0.96
	8	**301 K**	**421M**	**96%**	**0.96**	**0.96**	**0.96**
	10	377 K	515M	90%	0.90	0.91	0.89
	25	943 K	1,222M	77%	0.79	0.83	0.78
	50	1,887 K	2,445M	17%	0.16	0.11	0.08
	75	2,830 K	3,664M	6%	0.06	0.01	0.01

On the other hand, at a pruning rate of 10%, there are decreases of 4%, 3%, and 6% in accuracy values in rabbit, fowl, and all-species datasets, respectively. The results are consistent with other metrics such as recall, precision, and F1-scores, which perform in parallel with accuracy in these experiments.

Figure 6 illustrates the confusion matrices obtained from the three dataset classifications with both the pruned and original versions of the models. These confusion matrices pertain to the pruned models shown in Table 4, which demonstrate no reduction in accuracy for the datasets. Detailed examination of the confusion matrices demonstrates that the model's performance on the all-species dataset is similar to that on the other dataset. This observation is valid for both the pruned and original models. Even the pruned model performed enhanced classification proficiency for *Eimeria* species such as COE, MAG, VEJ, ACE, and MAX within the all-species dataset compared to the other ones. For instance, the classification precision for the COE species is $27/29 = 0.93$ in Figure 6b, while it is $146/154 = 0.95$ in Figure 6f. Similarly, the original model exhibits superior precision values for COE, INT, MAG, BRU, and MAX species within the all-species dataset.

Figure 7 visualizes the accuracy and loss values gathered by increasing the model's pruning rate across the datasets. Figure 7a. confirms that there is not a significant decrease in accuracy for all datasets up to a pruning rate of 10%. Beyond this threshold, a gradual decline in accuracy is observed. Interestingly, a significant peak appears within the range of 17–22%, followed by a subsequent decrease. Remarkably, the model operating on the all-species exhibits a performance of accuracy closely aligned with the other two datasets. Evaluating the results in terms of loss, Figure 7b shows that rabbit is the most negatively affected dataset by pruning.

4.3. Comparative Evaluation

In order to assess the efficacy of our pruning method, we conducted a comparative analysis of its performance against that of existing classification studies in the literature on both the fowl and rabbit versions of the *Eimeria* dataset. **Table 5** lists the studies in the literature, the dataset and classification method they employed, and the accuracy values they attained. Upon analysis of the studies on the fowl dataset, it is observed that the most successful result was achieved by an ANN model that employed CRF for feature extraction and the Relieff method for feature selection, with an accuracy of 96.6%. Our method, which utilized the pruned EfficientNetB0 model, attained the second-best result in the literature, with an accuracy of 95.1%. When examining

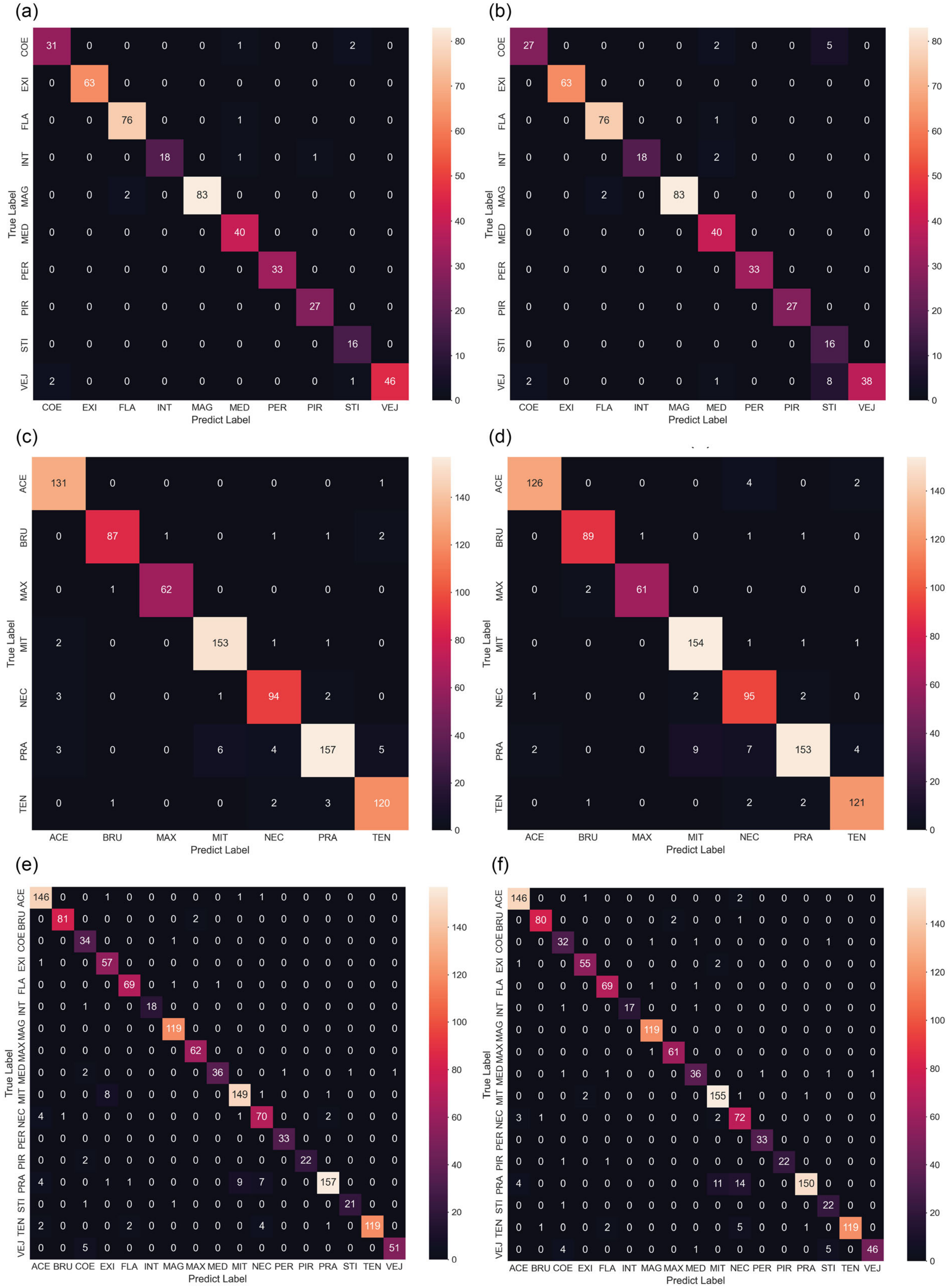

Figure 6. Confusion matrices of models' classification results: a) rabbit-original model, b) rabbit-pruned (5% pruning ratio) model, c) fowl-original model, d) fowl-pruned (7% pruning ratio) model, e) all-species-original model, and f) all-species-pruned (8% pruning ratio) model.

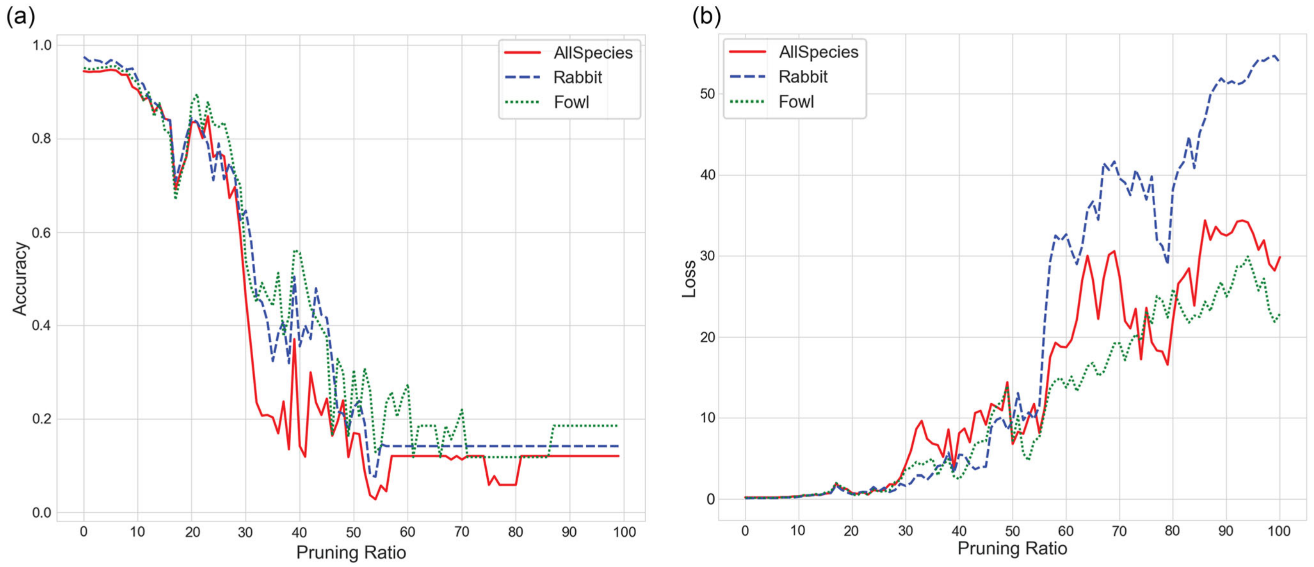

Figure 7. Performance alteration of the model regarding pruning rates for all datasets: a) accuracy and b) loss.

Table 5. The comparison of our pruned model with existing research in the literature on the classification of the *Eimeria* parasite. Bold show the performance of our method aganist the existing studies in the literature.

Authors	*Eimeria* type	Method	Accuracy [%]
Boufenar et al.[9]	Fowl	Xception	95.5
Buyukyilmaz et al.[35]	Fowl	MLP neural network	83.75
Diego et al.[37]	Fowl	CNN	93.12
Castanon et al.[31]	Fowl	Bayesian classifier	85.75
Abdalla et al.[34]	Fowl	ANN + CRF Feaute Extraction+Relieff Feature Selection	96.6
Our Pruned Model	**Fowl**	**EfficientNetB0**	**95.1**
Abdalla et al.[34]	Rabbit	ANN + CRF Feaute Extraction+Relieff Feature Selection	91.9
Abdalla et al.[33]	Rabbit	KNN	82.83
Our Pruned Model	**Rabbit**	**EfficientNetB0**	**97.4**

studies on the rabbit dataset, our method achieved an accuracy of 97.4%, surpassing the closest competitor[34] by 5.6%. These results underscore the superior classification performance of our method compared to existing approaches while also delivering energy savings. Overall, our findings highlight the practicality and efficacy of our proposed method.

5. Discussion

In this study, we initially identified the most proficient candidates of pretrained CNN models for our application domain by scanning 27 distinct extensively pretrained CNN models. This was achieved through a fine-tuning process focused on the classification layer, trained for five epochs. Our decision to set the epoch number at five is rooted in a twofold rationale. The first is to minimize energy consumption by reducing the resource-intensive nature of the fine-tuning process. The second is to identify a set of models that demonstrated rapid convergence toward their optimal performance within this short epoch range. Surprisingly, the selected models are the ones with the least number of parameters among the models in the pool of 27 pretrained models. When we carried out a more comprehensive performance analysis between these four models over 200 epochs, we found out that EfficientNetB0, which has the fewest parameters in our model pool, yielded the highest performance. This finding highlights the fact of overparameterization and redundancy in pretrained models. One of the important contributions of our study is that it saves significant energy by offering EfficientNetB0 to a practitioner who can use EfficientNetV2L, which has the highest number of parameters. EfficientNetB0 (4 million) has around 30 times fewer parameters than EfficientNetV2L (118 million) while also exhibiting superior performance in *Eimeria* parasite detection. In addition, by reducing the number of parameters with the pruning, we achieved an additional energy saving. Not updating the parameters in the convolution layer in the fine-tuning of the models has also contributed to this goal.

It is observed in the literature[9,34,37] that there is a common approach that has been taken to create two separate CNN models for the detection of rabbit and fowl parasites, separately. As this is mainly a parasite detection problem linked to microscopic images, a holistic approach has been taken to develop one single CNN model with the output of 17 classes and a similar number of CNN parameters in a single model. To clarify further, a single model for the rabbit had 10 output classes with a total number of 4 839 351 weights, and a single model for fowl had 7 output classes with a total number of 4 838 570 weights, whereas the combined all-species model with 17 output classes has resulted in a CNN model of 4 841 140 weights. While there is no significant reduction in the accuracy, there is a substantial potential gain for the energy consumption of the CNN model as the total number of weights used in the combined model is equivalent to that of a single model. Furthermore, this combined model has been developed as a template for parasite detection, which can be utilized as a foundation for the construction of other species' parasite detection models.

Our proposed pruning method removes 301 K unnecessary weight parameters on the EfficientNetB0 model for the all-species dataset. Despite this reduction, there is no decrease in the model's performance. In fact, as shown in the confusion matrices in Figure 6e,f, the pruned model was at least as successful as the original model in 9 out of 17 *Eimeria* species. This result proves that our proposed framework saves energy without losing accuracy with 8% pruning. On the other hand, Figure 7a reveals a different trend when the pruning ratio increases from 17 to 20, which leads to an accuracy enhancement. This observation emphasizes that all parameters in this range are unnecessary and even have a negative impact on the results. Consequently, the pruning rate ranges[1–8,17–20] in Figure 7a reveal the fact of overparameterization in the model.

As a result, the proposed methodology has successfully produced similar outcomes to those in the literature, even without updating the convolutional layer weights, which have a significant influence on the results of CNN models. It is evident that including the convolutional layer in the fine-tuning process would yield even greater success. In this direction, researchers aiming for higher accuracy in *Eimeria* classification can get superior results by conducting extended fine-tuning of EfficientNetB0. Although *Eimeria* parasites are the application area of our study, our proposed approach has the potential to be a role model for other fields.

A limitation of this study is that the model for the all-species dataset can only detect *Eimeria* species that cause disease in rabbits and fowls. While *Eimeria* parasites may cause significant diseases in a variety of animals, including cattle, sheep, and goats, the lack of publicly available data on these animals' *Eimeria* species has restricted our ability to develop a truly holistic model. By expanding the training dataset to include all *Eimeria* parasites from a broader range of hosts, it would be possible to create a more advanced model capable of detecting *Eimeria* infections across a wider spectrum of animal species.

6. Conclusion

Nowadays, high energy consumption and long running times are major challenges for pretrained CNN models. This is mainly due to the problem of overparameterization within these models. In this context, this study aims to mitigate the effects of the parameter redundancy drawback in these models by introducing a weight-level pruning method. Focusing on a specific application domain, the classification of *Eimeria* parasites, we performed a two-stage model selection procedure aimed at identifying the most energy-efficient pretrained model with superior classification performance among 27 cutting-edge candidates. The proposed framework determined the four most promising models, namely EfficientNetB0, EfficientNetV2B0, EfficientNetV2B3 and EfficientNetB1, in the initial stage and advanced to the next stage with them. In this second phase, with more extensive fine-tuning, EfficientNetB0 emerged as the model with the highest classification performance for *Eimeria* species. At the same time, the study evaluated the effects of the pruning technique on the model by utilizing the all-species dataset consisting of all *Eimeria* species.

As a result, our pruning method does not lead to a decrease in classification performance, even when reducing parameters by up to 8% across all *Eimeria* species. One of the remarkable results is that EfficientNetB0 has the least number of parameters among the 27 models. The modest parameter count of EfficientNetB0 corresponds to a significant reduction of the FLOPs in this model compared to its counterparts, resulting in EfficientNetB0 as the optimum choice for energy-efficient *Eimeria* parasite identification. Additionally, the performance of the holistic model running on the composite dataset encompassing all the species indicates that the proposed framework can be used as a general-purpose model for *Eimeria* parasite classification. This has resulted in a model with ≈50% fewer parameters than two separate models while maintaining nearly the same accuracy.

Considering that the developed approach reduces both energy consumption and complexity, it has a great potential to be applied to a broad spectrum of domains leading to the development of a universal *Eimeria* detection model that can be generalized and open to include other *Eimeria* parasites in one CNN model. Further work will be geared toward improving accuracy and generalization ability, pruning further, and adding more *Eimeria* parasites in a single model.

Acknowledgements

This work was supported by the Scientific and Technological Research Council of Türkiye (TÜBİTAK) under the BIDEB-2219 International Postdoctoral Research Fellowship Programme grant number 1059B192101063.

Conflict of Interest

The authors declare no conflict of interest.

Data Availability Statement

The data that support the findings of this study are openly available in The Eimeria Image Database at http://www.coccidia.icb.usp.br/imagedb/, reference number [38].

Keywords

convolutional neural networks, *Eimeria* classification, energy efficiency, green artificial intelligence, pretrained models, weight pruning

Received: October 8, 2023
Revised: May 3, 2024
Published online:

---

[1] M. Pezhman Pour, H. Seker, *Expert Syst. Appl.* **2020**, *144*, 113129.
[2] M. Sharif, G. Lűckemeyer, H. Seker, *IEEE Access* **2023**, *11*, 88583.
[3] C. Yao, W. Liu, W. Tang, J. Guo, S. Hu, Y. Lu, W. Jiang, in *Concurr. Comput.: Pract. Exp.* **2021**, *33*, e6064.

[4] M. Tan, Q. Le, in *Inter. Conf. on Machine Learning. PMLR*, **2021**, pp. 10096–10106.

[5] D. Moolchandani, A. Kumar, S. R. Sarangi,*J. Syst. Architect.* **2021**, *113*, 101887.

[6] R. Xie, X. Jia, L. Wang, K. Wu, *IEEE Wireless Commun.* **2019**, *26*, 161.

[7] Y. Tu, Y. Lin, *IEEE Access* **2019**, *7*, 58113.

[8] E. L. Denton, W. Zaremba, J. Bruna, Y. LeCun, R. Fergus, *Advances in Neural Inf. Process. Syst.* **2014**, *27, 1269*.

[9] C. Boufenar, S. Djamai, D. Mezghiche, L. Soudan, *2022 First International Conf. on Computer Communications and Intelligent Systems (I3CIS)*, IEEE, Piscataway, NJ **2022**, pp. 105–110.

[10] S.-K. Yeom, P. Seegerer, S. Lapuschkin, A. Binder, S. Wiedemann, K.-R. Müller, W. Samek, *Pattern Recognit.* **2021**, *115*, 107899.

[11] G. Hinton, O. Vinyals, J. Dean, Distilling the Knowledge in a Neural Network, arXiv:1503.02531, **2015**.

[12] W. Chen, L. Gao, X. Li, W. Shen, *Biomed. Signal Process. Control* **2022**, *71*, 103177.

[13] M. S. Khan, K. N. Alam, A. R. Dhruba, H. Zunair, N. Mohammed, *Comput. Biol. Med.* **2022**, *146*, 105581.

[14] P. Luo, Z. Zhu, Z. Liu, X. Wang, X. Tang, *Proc. AAAI Conf. Artif. Intell.* **2016**, *30*, 3560.

[15] S. Han, H. Mao, W. J. Dally Deep Compression: Compressing Deep Neural Networks with Pruning, Trained Quantization and Huffman Coding, *arXiv:1510.00149*, **2015**.

[16] Y. Gong, L. Liu, M. Yang, L. Bourdev, *Compressing Deep Convolutional Networks Using Vector Quantization. arXiv:1412.6115,* **2014**.

[17] Y. Guo, A. Yao, Y. Chen, *Adv. Neural Inf. Process. Syst.* **2016**, *29*, 1379.

[18] X. Jin, X. Yuan, J. Feng, S. Yan, *Training Skinny Deep Neural Networks with Iterative Hard Thresholding Methods. arXiv:1607.05423*, **2016**.

[19] S. Han, J. Pool, J. Tran, W. J. Dally, *Adv. Neural Inf. Process. Syst.* **2015**, *28*, 1135.

[20] V. Sindhwani, T. Sainath, S. Kumar, *Adv. Neural Inf. Process. Syst.* **2015**, *28*, 3088.

[21] A. Alqahtani, X. Xie, M. W. Jones, E. Essa, *Comput. Vis. Image Und.* **2021**, *208*, 103220.

[22] J.-H. Luo, J. Wu, An Entropy-Based Pruning Method for CNN Compression, arXiv:1706.05791, **2017**.

[23] M. Mondal, B. Das, S. D. Roy, P. Singh, B. Lall, S. D. Joshi, *Comput. Vis. Image Und.* **2022**, *222*, 103511.

[24] A. Kumar, A. M. Shaikh, Y. Li, H. Bilal, B. Yin, *Appl. Intell.* **2021**, *51*, 1152.

[25] S. Lin, R. Ji, C. Yan, B. Zhang, L. Cao, Q. Ye, F. Huang, D. Doermann, *Proc. of the IEEE/CVF Conf. on Computer Vision and Pattern Recognition*, IEEE, Piscataway, NJ **2019**, pp. 2790–2799.

[26] J.-H. Luo, H. Zhang, H.-Y. Zhou, C.-W. Xie, J. Wu, W. Lin, *IEEE Trans. Pattern Anal. Mach. Intell.* **2019**, *41*, 2525.

[27] M. Shao, J. Dai, J. Kuang, D. Meng, *Signal Image Video Process.* **2021**, *15*, 381.

[28] W. Zhang, Z. Wang, *Neurocomputing* **2022**, *512*, 40.

[29] M. Shao, J. Dai, R. Wang, J. Kuang, W. Zuo, *Int. J. Mach. Learn. Cybernet.* **2022**, *13*, 371.

[30] S. Ch, S. R. Dubey, M. Ghorai, *Neurocomputing* **2022**, *514*, 101.

[31] C. A. B. Castañón, J. S. Fraga, S. Fernandez, A. Gruber, L. Da F. Costa, *Pattern Recogn.* **2007**, *40*, 1899.

[32] S. Fernandez, A. H. Pagotto, M. M. Furtado, M. Katsuyama, A. M. B. N. Madeira, A. Gruber, *Parasitology* **2003**, *127*, 317.

[33] M. A. E. Abdalla, H. Seker, R. Jiang, in *2016 Inter. Conf. on Engineering & MIS (ICEMIS)*, IEEE, Piscataway, NJ **2016**, pp. 1–4.

[34] M. AE Abdalla, H. Seker, in *2017 39th Annual Inter. Conf. of the IEEE Engineering in Medicine and Biology Society (EMBC)*, IEEE, Piscataway, NJ **2017**, pp. 1517–1520.

[35] M. Buyukyilmaz, A. O. Cibikdiken, M. Abdalla, H. Seker, in *Proc. of the Inter. Conf. on Video and Image Processing*. **2017**, pp. 84–88.

[36] Z. Kucukkara, I. Ali Ozkan, S. Tasdemir, *Selcuk Univ. J. Eng. Sci.* **2022**, *21*, 69.

[37] D. F. Monge, C. A. Beltrán, **2019**.

[38] D. Liu, J. Yu, in *2009 Ninth Inter. Conf. on Hybrid Intelligent Systems*, IEEE, Piscataway, NJ **2009**, Vol. *1*, pp. 344–349.

[39] C. Zhang, P. Li, G. Sun, Y. Guan, B. Xiao, J. Cong, *Proc. of the 2015 ACM/SIGDA Inter. Symp. on Field-Programmable Gate Arrays* **2015**, pp. 161–170.

[40] M. Pakandl, *Folia Parasitol.* **2009**, *56*, 153.

[41] M. Tan, Q. Le, in *Inter. Conf. on Machine Learning. PMLR.* **2019**, pp. 6105–6114.

[42] G. Huang, Z. Liu, L. Van Der Maaten, K. Q. Weinberger, *Proc. of the IEEE Conf. on Computer Vision and Pattern Recognition*, IEEE, Piscataway, NJ **2017**, pp. 4700–4708.

[43] C. Szegedy, S. Ioffe, V. Vanhoucke, A. Alemi, *Proc. AAAI Conf. Artif. Intell.* **2017**, *31*, 4278.

[44] C. Szegedy, V. Vanhoucke, S. Ioffe, J. Shlens, Z. Wojna, *Proc. of the IEEE Conf. on Computer Vision and Pattern Recognition*, IEEE, Piscataway, NJ **2016**, pp. 2818–2826.

[45] A. G. Howard, M. Zhu, B. Chen, D. Kalenichenko, W. Wang, T. Weyand, M. Andreetto, H. Adam, *Mobilenets: Efficient Convolutional Neural Networks for Mobile Vision Applications. arXiv:1704.04861*, **2017**.

[46] M. Sandler, A. Howard, M. Zhu, A. Zhmoginov, L. Chen, *Proc. of the IEEE Conf. on Computer Vision and Pattern Recognition*, IEEE, Piscataway, NJ **2018**, pp. 4510–4520.

[47] B. Zoph, V. Vasudevan, J. Shlens, Q. Le, *Proc. of the IEEE Conf. on Computer Vision and Pattern Recognition,* IEEE, Piscataway, NJ **2018**, pp. 8697–8710.

[48] K. He, X. Zhang, S. Ren, J. Sun, *Computer Vision–ECCV 2016: 14th European Conf., Amsterdam, Netherlands, October 11–14, 2016, Proc., Part IV 14*, Springer, New York **2016**, pp. 630–645.

[49] F. Chollet, in *Proc. of the IEEE Conf. on Computer Vision and Pattern Recognition*, IEEE, Piscataway, NJ **2017**, pp. 1251–1258.