Final Report - Group H

Abstract

Animal species classification is the task of identifying and categorizing animals into their respective species. It is an important problem in the computer vision field and has many practical applications including wildlife monitoring, veterinary diagnosis, and biodiversity research. This is a challenging problem because there are a large number of different animal species, each with its own unique features, and many species may look similar. A deep learning model such as a convolutional neural network (CNN) can be trained on a large image dataset and the model learns to recognize patterns and features which are characteristic of each animal species. In our project, three datasets are selected from Kaggle consisting of different numbers of complexities. Three architectures named ResNet-18, MobileNet V2, and ShuffleNet V2 architectures were used to train models on each dataset resulting in accuracy between 30 % *to* 90 %. *Moreover, some pre-trained models were enhanced using transfer learning and hyperparameter tuning such as learning rate. Lastly, this report conducts the detailed analysis by comparing different architectures and datasets.*

A. Introduction

Wildlife monitoring, veterinary medicine, and zoology are areas of critical importance in real life. The correct identification of animal species is an essential aspect of these fields, and it is a challenging task due to the vast diversity of life on Earth, with over 8.7 million different species. Therefore, With the increasing availability of animal image datasets and advances in AI and Deep Learning techniques, the process of identifying species has gained attention from many researchers for the following reasons [1] [2].

Firstly, manual animal classification by human experts is time-consuming, labor-intensive, and prone to errors. Secondly, AI and Deep Learning-based animal classification can enable researchers to process large-scale datasets quickly and efficiently and Deep Learning models can provide high accuracy rates. This can help researchers and wildlife conservation organizations to monitor animal populations more effectively and identify trends or changes in population sizes, behavior, and habitats over time.

The primary challenge in animal image classification is to accurately recognize and distinguish between different species, breeds, and individual animals. Other challenges include handling variations in animal poses and capturing images in different environmental conditions. Additionally, the chosen datasets vary in terms of the number of classes, samples per class, and image quality, making it challenging to create a model that can generalize well across different datasets.

This project aims to study the performance of three different Convolutional Neural Network (CNN) architectures, namely ResNet18 [3], ShuffleNetV2 [4], and MobileNetV2 [5], on aforementioned three different animal datasets with two, ten and ninety classes. We will compare the performance of three models under the same training setup and hyperparameters to identify the most efficient model to classify animal species accurately. However, some models may perform well on one dataset but not on others, while others require a large amount of computational resources, such as high-end GPUs or cloud-based infrastructure. Moreover, some models may require extensive training time and data preprocessing to achieve optimal results. Thus, finding the most efficient model for a specific dataset remains a significant challenge [6]. To tackle these challenges, this project will use some data pre-processing techniques to address the noisy nature of the datasets. Transfer learning will be performed on two models with lower accuracy to compare their outcomes, and hyperparameter tuning will be performed on one of the noisy performing models to fine-tune it [7] [8] [9]. Evaluating the performance of models requires the use of several metrics such as accuracy, precision, recall, and F1 score.

It is important to note that every solution, regardless of how effective it may be, has its own set of pros and cons. When it comes to animal classification using deep learning, various techniques such as transfer learning, hyperparameter fine-tuning, and data augmentations have been recommended by researchers from different papers [10] [11] [12] [13] [14] [15]. Data augmentations, which is the part of data pre-processing are techniques that artificially generate new data by applying transformations such as rotation, scaling, and flipping to the existing dataset. Despite improving the accuracy and robustness of deep learning models by increasing the diversity of the training data, data augmentations may not always be suitable for certain animal classification tasks, and some augmentations may even introduce noise or artifacts to the dataset, which can negatively impact model performance [10] [11]. Transfer learning is a technique utilized to improve the performance of a new model on a different task including animal classification, where a large amount of labeled data is often not available to reduce the amount of data needed for the new task. However, pre-trained models may not always be suitable for the new task, and fine-tuning of hyperparameters may be necessary to achieve optimal performance [12]. Hyperparameter fine-tuning is another technique that has been recommended by researchers to improve the performance of deep learning models. Fine-tuning of hyperparameters involves adjusting these parameters, such as the learning rate, batch size, and the number of epochs to achieve the best performance for a specific task. However, fine-tuning requires a significant amount of computational resources and can be time-consuming [13].

The primary goal of this project is to perform a comparative analysis of all the models and find the most efficient model that can identify animal species accurately. To achieve this goal, we will implement and evaluate different CNN architectures using data pre-processing techniques, transfer learning, and hyperparameter tuning.

A.1. Related works

Krizhevsky et al. [16] were among the first to use CNNs in animal image classification. While this paper did not focus specifically on animal image classification, it introduced the AlexNet architecture, which was the first CNN to achieve state-of-the-art performance on the ImageNet dataset, which includes many animal categories. Many subsequent works have built upon the success of AlexNet and have proposed improvements to the CNN architecture for image classification tasks. For example, the VGGNet architecture proposed by Simonyan and Zisserman [17] in 2014 uses smaller convolutional filters and deeper network layers to achieve higher accuracy on the ILSVRC dataset. The Inception architecture proposed by Szegedy et al. [18] in 2015 further improves accuracy by using a combination of convolutional layers with different kernel sizes.

There have been several studies that have specifically addressed the classification of animal images. For example, Liu et al. [19] in 2018 proposed a CNN architecture based on the DenseNet architecture for classifying bird species. Similarly, Zhao et al. [20] in 2017 proposed a CNN architecture for classifying fish species using a combination of residual and inception blocks.

In addition to CNN architecture, many works have explored different approaches to data augmentation to improve image classification accuracy. For example, Zhang et al. [10] in 2019 proposed a method called mixup that generates new images by linearly interpolating between pairs of training examples. Zhou et al. [11] proposed a method called "RandAugment" that applies a random combination of image transformations, such as cropping, flipping, and color distortion, to create new training examples.

Transfer learning has been also widely used in animal image classification to leverage the pre-trained CNN models on large-scale datasets, such as ImageNet, to improve the classification performance on smaller-scale animal image datasets. For instance, Gao et al. [12] proposed a transfer learning-based approach for bird species classification, where they fine-tuned the pre-trained InceptionV3 model on the CUB-200-2011 dataset and achieved an accuracy of 95.21%. Similarly, in [13] a transfer learning-based approach has been proposed for dog breed classification, where they fine-tuned the pre-trained pre-trained InceptionV3, MobileNetV2, and NASNet. They retrain the fully connected layers on the Columbia Dog Dataset and achieved a maximum accuracy of 91%. In the following sections methodology of the approach and the results will be discussed.

Different regularization techniques have also been proposed to reduce overfitting, and optimization methods have been employed to improve the model's performance [14] [15].

B. Methodology

B.1. Datasets

The datasets utilized in this study were obtained from Kaggle, a popular platform for accessing publicly available datasets. A detailed description of the datasets is provided in this section.

The DogsAndCats dataset [21] contains images of only two classes, namely dogs and cats. The dataset consists of 10,000 JPEG images, with each class having 5,000 images. The distribution of images between the two classes is evenly balanced. Sample images from this dataset can be seen in Figure 1. Furthermore, Table 1 presents the statistical information of this dataset.

The Animal 90 dataset [22], on the other hand, comprises 90 classes, with an equal distribution of images among all classes. However, during data preprocessing, some classes were found to be empty or duplicated, resulting in the reduction of the number of classes to 90 and the number of images to 5,400 in total. This data-cleaning process was necessary to ensure the quality and accuracy of the dataset.

Lastly, the Animal 10 dataset [23] includes 10 classes with total image number of 26,000 JPEG images, but the distribution of images among these classes is not uniform, as illustrated in Figure 3. Figure 2 also displays sample images from this dataset, providing a visual representation of the data.

The datasets were preprocessed prior to feeding them into the model. The most important part of this phase was normalizing the images to smooth the process of training. Later, Augmentation techniques such as random rotation (degree=45), random horizontal flip, random vertical flip, and addition of Gaussian blur (sigma=(0.1,2)) were applied to enhance the diversity and variability of the data.

The entire dataset was initially split into train and test sets with a ratio of 8:2. Subsequently, the train set was further divided into train and validation sets using the same ratio, ensuring an appropriate distribution of data for model training and evaluation.

To ensure consistency, all images were resized to a resolution of 224x224. Upon analyzing the standard deviation of each dataset, it can be observed that the complex-

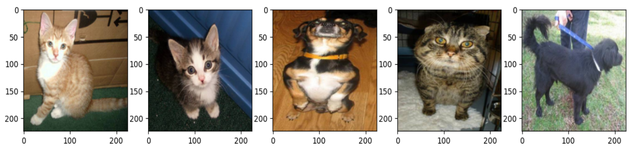

Figure 1. DogsAndCats sample images

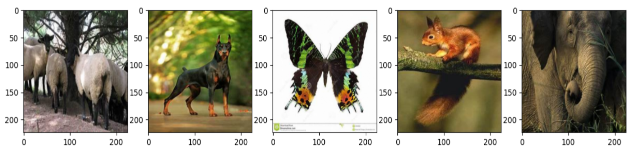

Figure 2. Animal10 sample images

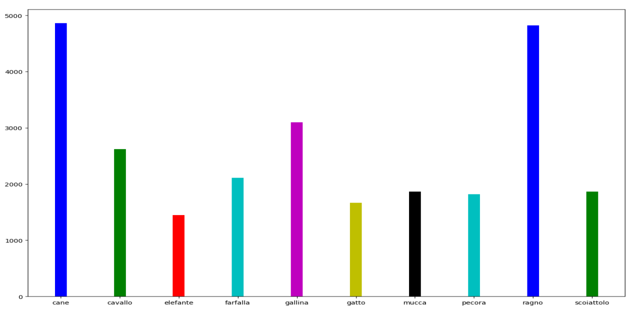

Figure 3. Class distribution of Animal10

ities of the datasets fall within a similar range. However, the DogsAndCats dataset exhibited a slightly higher level of complexity compared to the other datasets.

Table 1. Statistics of the Datasets

Attributes	Animal10	Animal90	DogsAndCats
Size	614MB	688MB	455MB
Number of Classes	10	90	2
Number of images	26000	5400	10000
Standard deviation	0.2139	0.2041	0.2226

B.2. CNN Models

Based on the nature of our datasets, which are relatively small compared to the massive Imagenet dataset with 14 million images, we have chosen to utilize popular deep neural network architectures including ShuffleNet V2, MobileNet V2, and ResNet-18 for our image classification tasks. These architectures differ in terms of their computational complexities and network structures.

Moreover, due to the limited availability of computational resources, we made a deliberate choice to adopt architectures that exhibit efficiency in terms of time complexity, striking a balance between computational cost and accuracy. However, it is worth mentioning that we also opted for one single structure that may be relatively computationally expensive in terms of complexity (ResNet-18).

ResNet18 introduced the concept of skip connections or residual connections. These skip connections enable faster training of deeper networks by mitigating the problem of vanishing gradients, which can occur in very deep networks. ShuffleNet V2 is known for its novel channel shuffling operation, which facilitates information exchange between channels, enhancing model performance. It also uses group convolutions to reduce computational complexity, allowing for efficient utilization of computational resources. MobileNet V2 is known for its depthwise separable convolutions, which decouple the spatial and channel-wise convolutions, significantly reducing the number of parameters and computations.

ShuffleNet V2 and MobileNet V2 are designed with efficiency in mind, featuring smaller model sizes and lower computational costs, making them well-suited for our resource-constrained environments. On the other hand, ResNet-18 is a deeper architecture with higher computational complexity, prioritizing accuracy over computational efficiency.

This distinction is evident when we compare the accuracy and time complexities of the three models, considering factors such as wall clock time for one epoch of training and the number of FLOP (floating point operations) calculations. As shown in Table 2, ResNet-18 requires more time for one epoch of training compared to ShuffleNet V2 and MobileNet V2. However, ResNet-18 also achieves higher accuracy compared to the other two models. This will be further discussed in section C of our report.

Table 2. Average time needed for one epoch of training

Attributes	Animal10	Animal90	DogsAndCats
ResNet-18	245s	117s	152s
ShuffleNet V2	214s	86s	127s
MobileNet	223s	90s	135s

B.3. Optimization Algorithm

During the model training process, an optimization algorithm was employed with the objective of minimizing the loss function. At each iteration, the CrossEntropyLoss function was used to calculate the loss value, which was then backpropagated through the network to update the weights towards minimizing the loss function. To assess the effectiveness of the optimization algorithm, a validation dataset was utilized, and the improvement was monitored in each epoch. The ADAM (Adaptive Moment Estimation) optimization algorithm was chosen for this task. ADAM computes adaptive learning rates for each parameter based on moving averages of past gradients and squared gradients, resulting in dynamic adjustments of learning rates during training. This allows for faster convergence and improved performance in deep learning models. In our implementation, we used an initial learning rate of 0.01. The hyperparameter tuning is discussed later in section C.3.

To evaluate the performance of the optimization model, we used metrics such as loss and accuracy values over the training and validation sets in each epoch. These metrics were plotted to visualize the model's performance after each epoch. In all cases, we observed an increase in accuracy and a decrease in loss values for both the training and validation sets, indicating the effectiveness of the Adam optimizer in minimizing the loss function and improving the model's performance. This validates the choice of Adam as the optimization algorithm for our image classification task. All of the plots are available in the result section of our GitHub page.

C. Results

C.1. Experiment setup

In this section, we outline the methodology employed in our experiment, including the setup, optimization, and validation of our model. We also detail the performance metrics utilized to evaluate the models, as well as the hyperparameters utilized in our experiment and their corresponding values.

The experiment was implemented in PyTorch, as it was one of the requirements for this study. Following data cleaning, the images were passed to the PyTorch dataset, and separate data loaders were created for each dataset. Three architectures, as mentioned in Section B.2, were trained on each dataset, and their performance on the training and validation datasets was plotted. To strike a balance between optimal training and computational resources, we utilized 50 epochs. A learning rate of 0.01, a commonly used value in image classification problems, was employed. Also, ADAM is chosen as the optimizer of the models as discussed in section B.3.

To evaluate the trained models, we utilized accuracy, precision, recall, and F1-measure on the test dataset. The result of the F1-measure which contains both precision and recall is provided in the table 4. Although high values for accuracy and F1-measure are desirable, there may be trade-offs among these metrics in certain cases. Therefore, it is important to consider these metrics collectively to obtain a comprehensive understanding of the classifier's performance.

C.2. Main Result

In this section, we analyze the results of each dataset and elucidate their behavior.

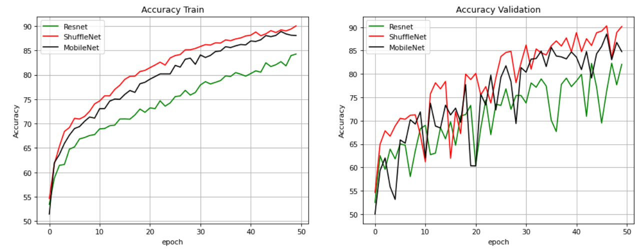

Figure 4. DogsAndCats accuracy on Train and Validation

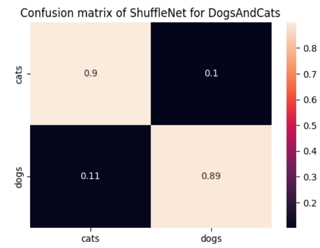

Figure 5. DogsAndCats and shuffleNet confusion matrix

C.2.1 Dataset Results

The results from the training and validation of DogsAndCats datasets are presented in Figure 4, where it is observed that the validation results exhibit considerable fluctuations. There are two possible explanations for this phenomenon. Firstly, the utilization of a high learning rate during training may contribute to the erratic behavior of the validation figures. A larger learning rate can cause the model to overshoot optimal weights, leading to instability and fluctuating performance. Secondly, the batch size employed during training can also be a significant factor in the observed fluctuations. Smaller batch sizes can introduce higher variance due to limited sample diversity, resulting in inconsistent validation results. By adopting a smaller learning rate and a larger batch size, it is expected that the fluctuations in the validation figures would be minimized, leading to more stable and reliable model performance. Also, the test accuracy for this dataset is provided in table 3. The highest accuracy among all models belongs to this dataset with the architecture of shuffleNet. This is because this dataset has only two classes and the classification problem can be solved rather easier than other datasets. The confusion matrix is also provided in figure 5.

In the Animal10 dataset, ResNet 18 achieved the highest accuracy on the training and test sets compared to the other models. This is likely due to ResNet's ability to mitigate the vanishing gradient problem, which can arise in deeper networks. The validation accuracy, however, was comparatively lower than the training and test accuracy, suggesting fine-tuning techniques may improve the gap and generalize the performance of the model accurately, which will be dis-

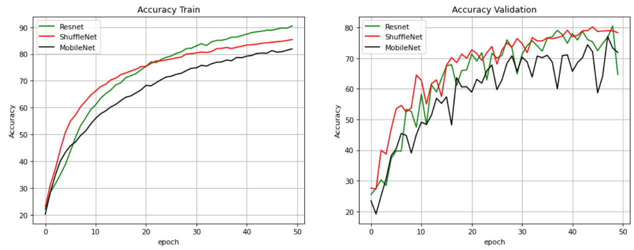

Figure 6. Animal 10 accuracy on Train and Validation

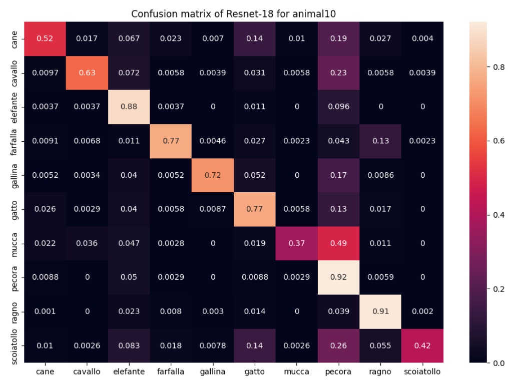

Figure 7. Resnet-18 confusion matrix for Animal 10

cussed in the Ablative section C.3. As you can see in Figure 3, the accuracies of the ShuffleNet V2 and MobileNet V2 models are lower than that of ResNet18, but they still achieved reasonable accuracies on the training set and the validation set that is demonstrated form Figure 6. This suggests that the model is able to learn the important features of the dataset, but not as well as the deeper ResNet18 architecture. Likely due to the ResNet18 depth and ability to learn complex features, it performs better on Animal10 which can be seen from the Confusion Matrix from Figure 7.

The results of training on the Animal90 dataset reveal that the three architectures have high training accuracy, but they perform poorly on the validation and test sets Table 3 and 8. These results suggest that the models are overfitting to the training set due to the limited number of images per class. Additionally, as you can see from Figure 8, the fluctuations and perturbations in the validation plots for all three architectures suggest that the models may be sensitive to the choice of learning rate or batch size. It may be beneficial to explore transfer learning techniques, where pre-trained models on larger datasets are fine-tuned on the Animal90 dataset, as this can help improve performance with limited training data, which we discuss in section C2.4.

C.2.2 Transfer Learning

Based on the observed performance of the models on the Animal90 dataset, we conducted a study to investigate the impact of transfer learning using ShuffleNet and MobileNet architectures. The results, as depicted in Figure 9, demonstrated significantly improved accuracy in the training and validation datasets. Specifically, the test accuracy for MobileNet was 49.21%, while that for ShuffleNet was 66.01% with transfer learning, indicating the effectiveness of utilizing pre-trained weights for better generalization on the same dataset.

These outcomes highlight the importance of weight initialization during the training process, as evidenced by the superior performance of the models with transfer learning. However, it should be noted that overfitting still persisted in this dataset, likely due to the limited number of images per class and the high number of classes in the Animal90 dataset.

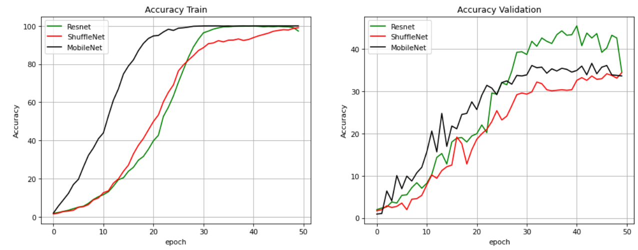

Figure 8. Animal 90 accuracy on Train and Validation

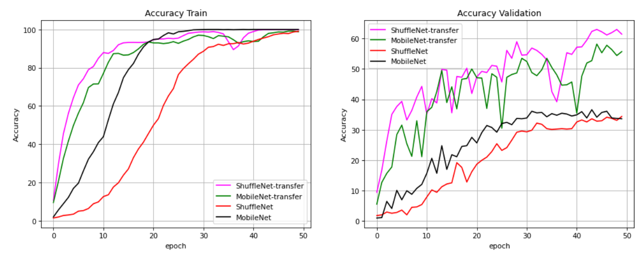

Figure 9. Accuracy of Transfer Learning models compared to none transfer learning models

Table 3. Test Accuracy of the nine models

Attributes	Animal10	Animal90	DogsAndCats
ResNet-18	79.43%	41.01%	80.88%
ShuffleNet V2	77.87%	36.81%	89.63%
MobileNet V2	72.20%	33.69%	85.15%

Table 4. F1 Score of the nine models

Attributes	Animal10	Animal90	DogsAndCats
ResNet-18	76.11%	41.01%	80.57%
ShuffleNet V2	76.37%	36.81%	89.51%
MobileNet V2	75.25%	33.69%	85.14%

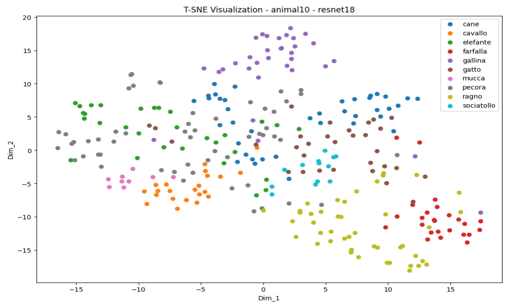

Figure 10. tsne - animal10 - resnet

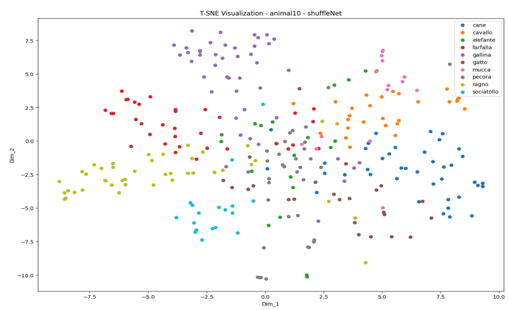

Figure 11. tsne - animal10 - shuffleNet

Table 5. Hyperparam tuning - validation accuracy after 10 epochs

Learning rate	0.1	0.01	0.001	0.0001	0.005	0.0005
accuracy	18.99%	59.54%	59.91%	37.32%	61.84%	57.49%

C.2.3 T-SNE

As part of our research, we conducted a study to assess the efficacy of different models in segregating data points after the final fully connected layer. This investigation was carried out using two datasets, DogsAndCats and Animal10, and two distinct architectures, Resnet and ShuffleNet. A total of 300 data points were analyzed, and the findings are presented in Figures 10, 11, 12, and 13. The results indicate that the models performed remarkably well in discriminating the data points in the DogsAndCats dataset. However, their performance was comparatively less satisfactory in the Animal10 dataset, which can be attributed to the higher number of classes and data points in the Animal10 dataset, posing greater complexity and challenges for accurate segregation.

C.3. Ablative Study

We conducted hyperparameter tuning on a specific dataset and architecture, focusing on a single hyperparameter - the learning rate. Based on the observed results, it was evident that the model's performance was influenced

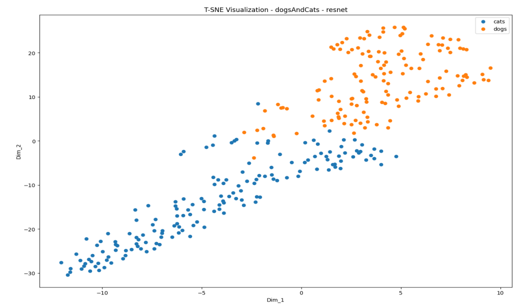

Figure 12. tsne - dogsAndCats - resnet

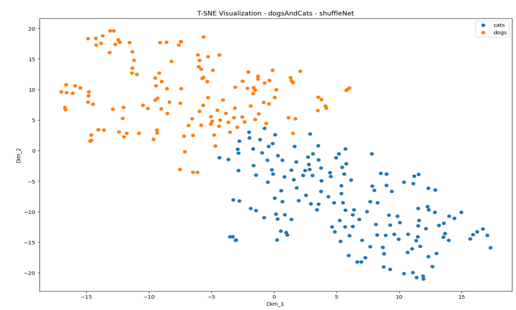

Figure 13. tsne - dogsAndCats - shuffleNet

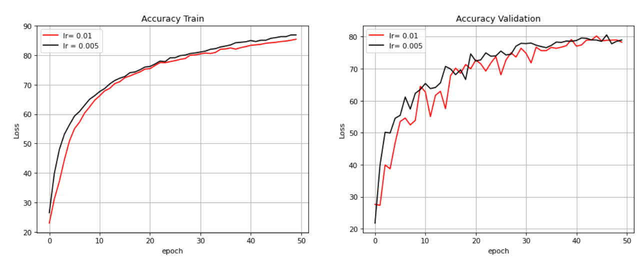

Figure 14. Accuracy on Train and validation for the old and new learning rate

by the complexity and size of the dataset. Consequently, we proceeded to train the ShuffleNet V2 model on the Animal10 dataset with varying learning rates, ranging from 0.1 to 0.0005, and evaluated the accuracy after 10 epochs. The findings, presented in table 5, revealed that the model trained with a learning rate of 0.005 exhibited slightly higher accuracy on the validation set, with reduced fluctuations and perturbations in the validation plot. This suggests that a lower learning rate could potentially aid in better convergence and improved generalization of new data. Figure 14 illustrates the difference in accuracy between the training and validation sets for the two learning rates. Notably, the test accuracy achieved with the optimized hyperparameters was 78.43%, reflecting a 1% improvement compared to the previous case.

References

[1] Clement Duhart, Gershon Dublon, Brian Mayton, Glorianna Davenport, and Joseph A Paradiso. Deep learning for wildlife conservation and restoration efforts. In *36th International Conference on Machine Learning, Long Beach*, volume 5. PMLR, 2019. 1

[2] Brenda McComb, Benjamin Zuckerberg, David Vesely, and Christopher Jordan. *Monitoring animal populations and their habitats: a practitioner's guide*. CRC Press, 2010. 1

[3] Kaiming He, Xiangyu Zhang, Shaoqing Ren, and Jian Sun. Deep residual learning for image recognition. In *Proceedings of the IEEE conference on computer vision and pattern recognition*, pages 770–778, 2016. 1

[4] Ningning Ma, Xiangyu Zhang, Hai-Tao Zheng, and Jian Sun. Shufflenet v2: Practical guidelines for efficient cnn architecture design. In *Proceedings of the European conference on computer vision (ECCV)*, pages 116–131, 2018. 1

[5] Mark Sandler, Andrew Howard, Menglong Zhu, Andrey Zhmoginov, and Liang-Chieh Chen. Mobilenetv2: Inverted residuals and linear bottlenecks. In *Proceedings of the IEEE conference on computer vision and pattern recognition*, pages 4510–4520, 2018. 1

[6] Jwan Saeed and Subhi Zeebaree. Skin lesion classification based on deep convolutional neural networks architectures. *Journal of Applied Science and Technology Trends*, 2(1):41–51, 2021. 1

[7] Mingsheng Long, Yue Cao, Jianmin Wang, and Michael Jordan. Learning transferable features with deep adaptation networks. In *International conference on machine learning*, pages 97–105. PMLR, 2015. 1

[8] Romain Mormont, Pierre Geurts, and Raphaël Marée. Comparison of deep transfer learning strategies for digital pathology. In *Proceedings of the IEEE conference on computer vision and pattern recognition workshops*, pages 2262–2271, 2018. 1

[9] Andrés Ovidio Restrepo Rodríguez, Daniel Esteban Casas Mateus, Paulo Alonso Gaona García, Carlos Enrique Montenegro Marín, and Rubén González Crespo. Hyperparameter optimization for image recognition over an ar-sandbox based on convolutional neural networks applying a previous phase of segmentation by color–space. *Symmetry*, 10(12):743, 2018. 1

[10] Hongyi Zhang, Moustapha Cisse, and Yann N Dauphin. mixup: Beyond empirical risk minimization. *arXiv preprint arXiv:1710.09412*, 2018. 1, 2

[11] Ekin D Zhou, Hanting Fan, Yiming Cao, Ayad Erraqabi, Yiming Wang, Po-Yao Sheu, Yuwen Zhang, Haoran Bai, Shuyang Bai, and Jianbo Ye. Randaugment: Practical automated data augmentation with a reduced search space. *arXiv preprint arXiv:1909.13719*, 2020. 1, 2

[12] Mohammed Alswaitti, Liao Zihao, Waleed Alomoush, Ayat Alrosan, and Khalid Alissa. Effective classification of birds' species based on transfer learning. *International Journal of Electrical and Computer Engineering (IJECE)*, 12(4):4172–4184, 2022. 1, 2

[13] Punyanuch Borwarnginn, Kittikhun Thongkanchorn, Sarattha Kanchanapreechakorn, and Worapan Kusakunniran. Breakthrough conventional based approach for dog breed classification using cnn with transfer learning. In *2019 11th International Conference on Information Technology and Electrical Engineering (ICITEE)*, pages 1–5. IEEE, 2019. 1, 2

[14] N Manohar, YH Sharath Kumar, Radhika Rani, and G Hemantha Kumar. Convolutional neural network with svm for classification of animal images. In *Emerging Research in Electronics, Computer Science and Technology: Proceedings of International Conference, ICERECT 2018*, pages 527–537. Springer, 2019. 1, 2

[15] S Suryawanshi, V Jogdande, and A Mane. Animal classification using deep learning. *Int. J. Eng. Appl. Sci. Technol*, 4(11):305–307, 2020. 1, 2

[16] Alex Krizhevsky, Ilya Sutskever, and Geoffrey E Hinton. Imagenet classification with deep convolutional neural networks. *Communications of the ACM*, 60(6):84–90, 2017. 2

[17] Karen Simonyan and Andrew Zisserman. Very deep convolutional networks for large-scale image recognition. *arXiv preprint arXiv:1409.1556*, 2014. 2

[18] Christian Szegedy, Vincent Vanhoucke, Sergey Ioffe, Jon Shlens, and Zbigniew Wojna. Rethinking the inception architecture for computer vision. In *Proceedings of the IEEE conference on computer vision and pattern recognition*, pages 2818–2826, 2016. 2

[19] Wei Liu, Kai H Ang, and Jun Zhang. Deep learning-based bird species classification using color histograms and densenet. *Ecological Informatics*, 47:40–47, 2018. 2

[20] Jianwei Zhao and Qinzhang Wu. Fish species recognition based on convolutional neural network with residual and inception blocks. *Neurocomputing*, 267:217–223, 2017. 2

[21] chetanimravan. Dogs cats images. https://www.kaggle.com/datasets/chetankv/dogs-cats-images. 2

[22] Sourav Banerjee. 90 animal species. https://www.kaggle.com/datasets/iamsouravbanerjee/animal-image-dataset-90-different-animals. 2

[23] Corrado Alessio. Animals-10. https://www.kaggle.com/datasets/alessiocorrado99/animals10. 2