COMP 6721 (Applied Artificial Intelligence) – Project Report

Breed Identification System (Group B)

Abstract

The absence of information about available breeds is currently the most serious obstacle to understanding and conservation of various species. This report introduces Breed Identification System which is primarily focused on identifying the correct class(breed) of a given species. For the scope of this project, we are using 3 different data sets and 3 different CNN architectures. We took three datasets dog, cat, and fish with varying number of classes and images in each dataset. All datasets were trained on 3 CNN architectures MobileNetV2, ResNet18 and ResNet50, thereby resulting in 9 instances from scratch, 6 instances for transfer learning and multiple instances for hyperparameter tuning. In this report we discuss different outcomes we got from training (scratch and transfer), hyperparameter tuning, TSNE visualization and helps us understand which model-dataset combination provides maximum accuracies along with optimal hyperparameters for training them.

1. Introduction

Breed Identification System is designed to classify the breed-class of an animal. With scientific advancement, curiosity has also grown prompting humans to learn more about different species as within the same species there can be numerous breeds. This is difficult to distinguish with the naked eye considering the number of breeds that exist for any animal. Deep learning [1] can be used to solve this issue. Computer Vision models [2] can be built to aid this, we plan on using a combination of scratch training and transfer learning using state-of-the-art CNN models [3] to build a solution of the same. Practical applications of our system include pet stores, vet clinics, endangered species survey, potential ecological background study of breeds, etc.

Challenges: Differences in images used in the dataset, featuring animals of the same breed in a variety of lightings and positions, which is a problem. The other issue is the lack of balanced datasets which can have hit on overall performance. Similar challenges have been faced by other breed identification solutions as well. One of the papers that was referred for this project 'Animal Species Recognition Using Deep Learning' [4] in which they used a dataset furnished by BCMOTI. This dataset was small (50k images) and imbalanced. To fix imbalance we augmented the dataset by using a different dataset 'Snapshot Wisconsin' which has high resolution images (between 2048 × 1536 and 512 × 384 pixels) and 0.5 million captured events. This is a good way to solve the issue but can also add additional issues if a good dataset is not found. Also, we were specifically instructed to stick to the datasets we have chosen. Unbalanced dataset for dog and cat was a major issue during the implementation phase while training from scratch for the 9 model instances. In this project we try to solve this issue by data augmentation and removing some classes with less images in each dataset, so that we have a relatively similar number of images. Normalization [5] was done on the data which helped significantly in increasing the accuracy. Through this study we attempt to explore and provide detailed analysis (based on our evaluation metrics) of how different CNN architectures fare against the chosen datasets of animals. Dog, cat, and fish are the three datasets used and MobileNetV2, ResNet18 and ResNet50 are the CNN architectures on which datasets are trained. 9 instances trained from scratch were achieved after data preprocessing, 6 instances of transfer learning for dog and cat dataset and multiple instances for hyperparameter tuning [6] of learning rate and loss function. We have done a thorough comparison and point out appropriate models through which high accuracy can be achieved. The visualization of data has been done using t-SNE, which is used particularly for high-dimensional data and functions by assigning each datapoint a location in a two or three-dimensional map. We believe any improvements we make in this project can be ported to other animal breed identification without much changes through transfer learning.

1.1. Related work

In this section, we discuss the existing approaches to the problem of breed classification as well as briefly discuss the CNN models we plan to use.

Many initiatives have been taken to identify animals using technology by using pictures of different breeds; however, most have been undertaken on a single type of animal, such as a dog, limiting its use to other species of animals. The studies done on stable datasets are limited by the availability of the number of classes and quantity images in them. [7]

In 'Animal Recognition and Identification with Deep Convolutional Neural Networks for Automated Wildlife Monitoring' [8] paper authors focused on wild animal recognition using trap camera images. They discussed issues with image capturing wherein in the majority of images, the animal is partially captured or not captured at all. Dataset used in this paper is obtained through the Wildlife Spotter project, which is a project undertaken by Australian organizations and universities to classify animals. Scientists are helped by volunteers called 'Citizen Scientists' in classification which results in creation of a great dataset. 3 models were used Lite AlexNet, VGG-16 and ResNet-50, but detection was limited to bird, rat, and bandicoot (and no detection of various breeds within a species).

Furthermore, there were studies in which the facial attributes were stored as vectors. The pictures being evaluated had a little weight variation from the train images, thus they were classified as a breed. If an image of a dog is given, [9] it works to determine the dog's breed and corresponding attributes; otherwise, it tries to find facial features that exist in a dog and vice versa. Another paper that worked on this problem proposed Part Localization for the categorization of dog breeds. In paper [10] the authors Liu, J. Kanazawa and A., Jacobs developed an exemplar-based geometry and appearance models of dog breeds and their facial components to identify precise predictions. However, in the implemented model trial, they only had a detection accuracy of 67Apart from working on the dog datasets, some papers also studied the implementation of a deep convolutional neural network (CNN) on other species as well. In [11], the authors Guobin Chen and Tony X. Han have attempted to classify 20 animal species on a dataset of their own, containing 20,000 images. The animals were cropped out from the images using an automatic segmentation method (ensemble video object cut), and these crops were then used to train and evaluate their system. The accuracy for this was 38.31% [12] which is very low compared to the other models. In [14] R. Zhang compares performance of different deep learning models, like VGGNet, Inception-v3 and the optimized deep learning model in cat breed recognition and obtains an accuracy of about 84

2. Methodology

2.1. Datasets

Dog dataset is canine subset created by Fei-Fei Li et al [1] of Stanford in order to practice fine-grained image categorization released in 2011. Cat Dataset was made available by PetFinder API using the steps of [13]. Fish dataset was created to carry out segmentation, feature extraction, and classification tasks by Ulucan and. Karakaya et al [14] of Izmir University, Turkey.

Animal	Number of Images	Image sizes	No. of breeds
Dog [15]	20.6 k	{400 - 500} X {310 - 345}	120
Cat [16]	127 k	{300 - 330} X {250 - 270}	67
Fish [17]	9 k	590 X 445	9

Preprocessing of the data was done iteratively initial transformations of the images was done using torchvision transform modules before breaking them into: train (70% of total images per class), validation (10% of total images per class), and test (20% of total images per class) and shuffled the images to prevent overfitting/underfitting on the same samples.

The fish dataset was evenly balanced with 1000 images per class (9k total). Whereas, in the dog dataset some classes had less than 75 images, so we eliminated them and reduced the dataset to 50 breeds with a total dropping to 9719 images The Cat dataset was initially pruned, reducing the number of images to 35k from 127k and the number of classes to 27 from 67 in the original dataset. We chose a cut-off of 500 images per class out of fairness for the model's learning abilities and through rounds of training. So, classes for Cat dataset ranged from 500 3500 images.

For pre-processing, we generated the mean and stand deviation values for dog and cat dataset in order to normalize them. Apart from normalization we also did some basic transform operations like,

- Resizing the image to a median crop value (though not explicitly required by the model).
- Taking a random crop anywhere within the image of 224*224.
- horizontally flipping random images based on a probability factor of 0.5.

2.2. CNN Models

In terms of architectures [4], we are using ResNet18, ResNet50 and MobileNetV2 and have created 9 instances as train from scratch (considering 3 datasets on each of the three CNN models). We also have 3 instances of transfer-learning the dog dataset. So, there are a total of 12 model instances whose results are discussed below. The reason for choosing these three models is based on our time-taken vs performance analysis of the popular CNN architectures trained for object identification/classification. The ResNet18 model has 72 layers with 18 deep layers and ResNet50 is a CNN with 50 deep layers. However, ResNet50 outperformed the transfer learning method because this pretrained network can classify images into 1000 object categories, including animals, which is relevant to our problem.

MobileNetV2, which is a convolutional neural network with 53 deep layers. Being the lightest of 3, naturally it gave mixed results in scratch training but was able to achieve good performance via transfer learning. We have maintained the following hyperparameters across all models.

Another reason for choosing ResNet18 is it has faster execution comparing with other variants of ResNet and it helps in maintaining a low error rate much deeper in network with only 240M Flops [18]. The positive side of Resnet50 is that it uses bottleneck design for building block [A bottleneck residual block uses 1x1 convolutions, which reduces the number of parameters and matrix multiplications].

The reason behind opting MobilenetV2 is because of channel reduction implementation means channels number in the 1x1 convolutions divided by 6 and only the depth wise convolution uses this high number of channels, not the 1x1 convolutions comparing with other variants of Mobilenet.

Talking about computational complexities of all 3 models for training and validation in terms of wall clock time for one-epoch is as follows:

		Dog	Fish	Cat
Train From Scratch	ResNet-18	160	100	500
	ResNet-50	254	148	790
	MobilenetV2	170	120	480
Transfer Learning	ResNet-18	128	117	-
	ResNet-50	146	112	-
	MobilenetV2	122	100	-

Considering about Flops for each 3 CNN architecture models, ResNet18 model has around 2 billion Flops, ResNet50 model has around 3.8 – 4 billion Flops and MobilenetV2 model has 1.5 billion Flops.

2.3. Optimization Alogrithm

For evaluating of the model performance at every epoch we have segregated 10% of the available data as Validation and 20% as test (hold-out) data upon which all evaluation metrics was calculated and presented in this report.

Adam [19] is the optimization algorithm of choice for all 9 dataset-model (instance). We introduce Adam, an algorithm for first-order gradient-based optimization of stochastic objective functions, based on adaptive estimates of lower-order moments. The method is computationally efficient, has little memory requirements, is invariant to diagonal rescaling of the gradients, and is well suited for problems that are large in terms of data and/or parameters. [11].

An additional benefit of using Adam optimizer is it's faster and rectifies vanishing learning rate but not without having an inverse effect on computational cost at the same time.

For evaluation we relied on the popular scikit-learn library to generate a classification report comprising of accuracy, precision, recall and F1 score. We also included the ROC_AUC score for each class of the dataset.

Hyperparameters which are common across all codes are as follows:
Optimizer $\rightarrow$ Adam , Loss $\rightarrow$ Cross Entropy Loss, Learning rate $\rightarrow$ 0.001 , Batch size $\rightarrow$ 64 We talked about our choice of Adam optimizer above, but for other parameters it's merely about depth of knowledge and minimizing compute times (batch size chosen to maximize while also not running into out of memory CUDA errors).

3. Results

3.1. Experiment Setup

One of the core requirements of training large NNs is compute resources, our initially setup was to run in our local machines but due to severe GPU limitations and time constraint we moved over to cloud service providers like Kaggle, Colab etc. and through our experimentation and analysis we found Kaggle to be better suited in the long run, since it provided faster GPUs and allowed us to run overnight training without interruptions.

Optimization for our models involved tuning at various levels,

I **Data:** Pruning the dataset to deal with class imbalance, and only selecting classes with a minimum number of images to dropping problematic classes which had highly irregular images or lack thereof. (3.2)

II **Normalization:** We homogenized the Cat and Dog datasets to bring order and limit the range of values in hopes of helping the model learn better features. We began this by using ImageNet's standard deviation and mean values for normalization, but the results were unsatisfactory achieving a mere 30% accuracy. Eventually, we generated the normalized values according to the dataset and it helped boost the performance (results of which is discussed in 3.2)

III **Hyperparameter Tuning:** Assessing the effects of various hyper parameters (refer 3.3 for more details) like loss functions, learning rate etc. [20]

The datasets were broken down into 70/10/20 splits with 20 percent of the original images serving as the test(hold-out) data which was used to compute various evaluation metrics (Table 2).

3.2. Main Results

	ResNet - 18			ResNet - 50			MobileNetV2		
	D	F	C	D	F	C	D	F	C
Accuracy (%)	69	99	58	69	91	58	64	81	63
Precision (%)	48	99	50	60	92	54	65	85	69
Recall (%)	41	99	49	56	91	51	62	81	51
F-1score (%)	41	99	49	56	91	55	63	80	69
Time Taken per epoch	100	100	500	120	148	784	90	120	477

We have achieved good results across all evaluation metrics on the Fish dataset as seen in Table 1. Resnet18 seems to be the best suited model maxing out at 99% accuracy. Resnet-50 had lesser fluctuations in validation loss though, but they all converged around the same optimal value. Our initial experimental for fish without normalization provided accuracies above 90%, so we didn't foresee the need to include normalization. Unfortunately for the dog dataset, we obtained low accuracies across all models while training from scratch during our first attempt due to improper image distribution among breeds and a smaller number of training samples for each class. Later, the dataset was pruned, and an attempt was made to use standard normalized values (ImageNet) but only achieved 30% accuracy. Finally, during the third iteration, the normalized values were generated for the dataset and then we merged the validation train images instead of our standard 70/10/20 split to increase the train sample size hence the lack of validation plots in Fig. 2. Accuracies ranging from 58-64% was observed which is a huge leap compared to the previous two trials.

Cat dataset, similar pruning/dropping methods were followed where the number of classes were reduced from 67 $\rightarrow$ 55 $\rightarrow$ 27. In our last cut any class below 500 samples were dropped to improve performance. For our latest iteration we used normalized values generated for the dataset. Accuracies started at 15% and through our trails have reached a meteoric 58% in our last run. For evaluation, we went ahead with the built functions of scikit-learn library (classification report). We also calculated class-wise accuracies to understand the discrepancies in evaluation between classes, which helped us understand are generally underfitting when compared to others. (In Cat dataset American Bobtail: 13.23%, American Shorthair: 21.65% these two classes were failing to achieve good acc even after 100s of epochs of training) which raises the question about the quality of images in a few breeds (future improvements). The hyper-parameters [5] like learning rate, batch-size, or optimizer function for transfer learning on cat dataset will be modified to get better results (Ablation Study).

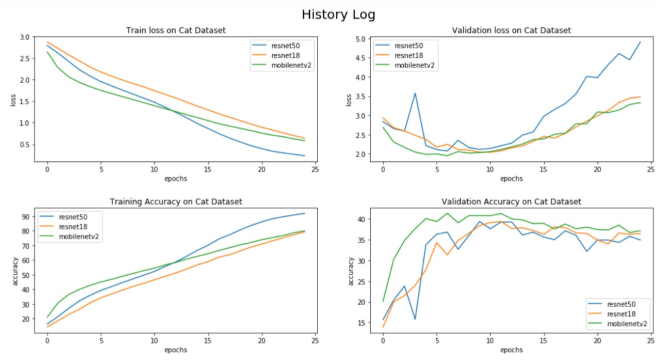

Figure 1. Loss and Accuracy plots for Cat.

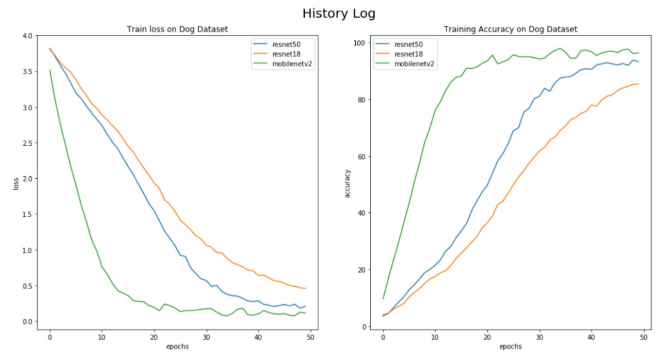

Figure 2. Loss and Accuracy plots for Dog.

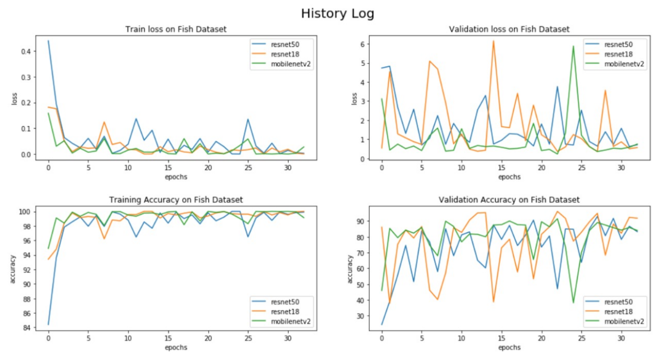

Figure 3. Loss and Accuracy plots for Fish.

Results Obtained when model weights were not random (Transfer Learning).

D $\rightarrow$ Dog, F $\rightarrow$ Fish

	ResNet-18		ResNet-50		MobileNetV2	
	D	F	D	F	D	F
Accuracy (%)	77	98	83	97	80	99
Precision (%)	78	99	84	98	81	100
Recall (%)	77	99	83	98	80	100
F-1 Score(%)	76	99	82	98	80	100
Time/ epoch (seconds)	128	100	146	112	122	99

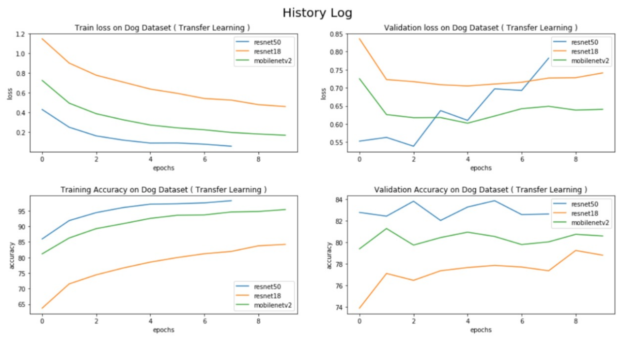

Figure 4. Loss and Accuracy plots for Dog(TL).

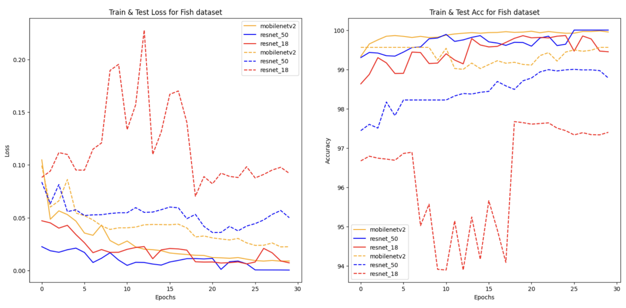

Figure 5. Loss and Accuracy plots for Fish (TL).

But the actual importance of transfer learning can be shown for dog model as the improved accuracy was around 80% from 60% (train from scratch model instance) which shows that training CNNs from learned weights (transfer learning) can help achieve better results quicker than training from scratch. Runtime for ResNet18 was generally comparable to Mobilenetv2 but results across various evaluation metrics prove that ResNet18 is better suited for animal-breed prediction.

Visualization of features based on model predictions using T-SNE align with our evaluation prompting a clear distinction in the boundary for the Fish dataset (both trained from scratch and transfer learning Fig 6 - 9.) whilst showing unclear weaved separation for the dog dataset train from scratch and TL as well prompting the model was not able to learn distinguishable features for a large set of samples which makes it even more difficult to represent them in lower dimensions (PCA) through T-SNE visualization. We used a scatter plot throughout visualization and numerically labelled the classes to make them presentable. You can find the plots in Appendix section.

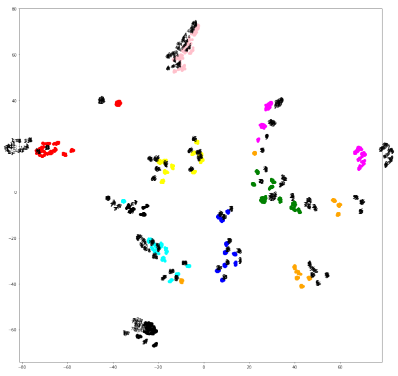
Figure 6. Scatter plot of TSNE features of Fish-ResNet18.

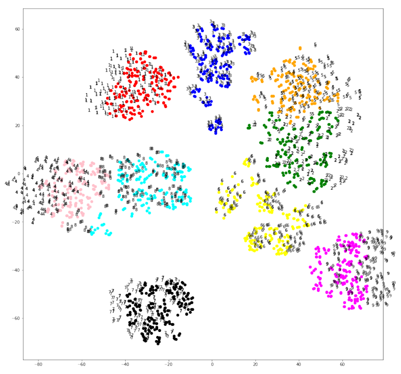
Figure 7. Scatter plot of TSNE features of Fish-ResNet50 (TL)

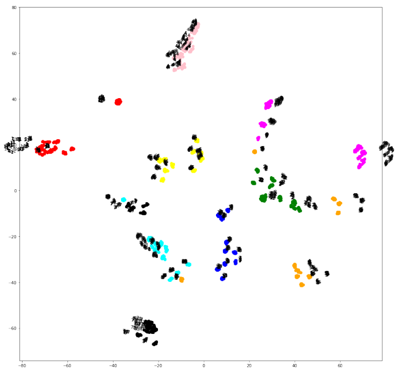
Figure 8. Scatter plot of TSNE features of Fish-ResNet50.

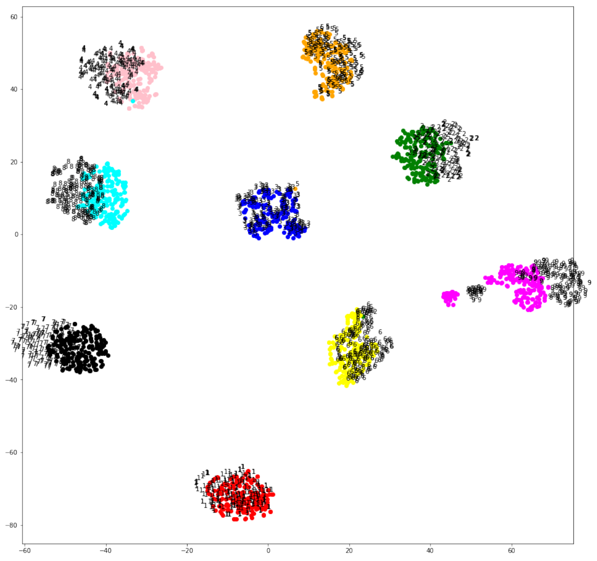

Figure 9. Scatter plot of TSNE features of Fish-ResNet50.

3.3. Ablative Study

In an attempt to better understand the effects of different hyperparameters we began varying them within the scope of the argument to study them. Our first choice for this experiment was Loss functions [21] but quickly ran into a halt when the appropriate list of choices dwindled down to three i.e., MultiMarginLoss, NLLLoss and CrossEntropyLoss (original loss function) , results of this short study prompted that our original loss function was the best choice performing better in all evaluation metrics and accuracy beings to falter when modified.

LR	.001	.01	.005	**.0015**	.00075
Accuracy	58	36	32	62	54
Precision	50	28	26	60	52
Recall	49	31	28	61	51
F-1 Score	49	29	27	60	51

To conduct a thorough study and to satisfy the requirements of the project, we chose learning rate as a hyper parameter since it gives us unabridged control. [22] Our chosen values for this experiment is [0.01, 0.001, 0.0015, 0.00075, 0.005],

Based on the above [23] table we can draw the following interpretations,

I Learning rate 0.01 (10x standard value) has a 20 percent margin in accuracy indicating that the step size might have been too large, and it could have missed converging on the minima.

II Learning rate 0.005 (5x standard value) also showcased a similar drop in accuracy further solidifying our inference about the gradients leaping over the local minima.

III Learning rate 0.00075 (0.75x standard value) gave a 54% accuracy (closer to our reported result for the cat-Resnet18 combination) prompting similar convergence to your standard model.

IV Learning rate 0.0015 (1.5x standard value) gave the highest accuracy out of our experiments reaching 62.8%

V Learning rate 0.001 (standard value) was our original hyper parameter value which we used across all instances.

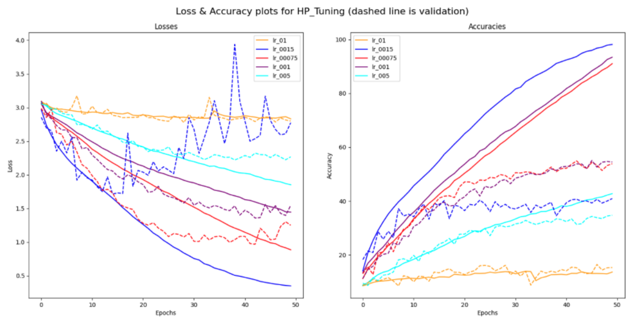

Figure 10. Loss and Accuracy plots for HP Tuning.

Train and Validation plots for our HP tuning experiment can be found in the Fig 6. Based on both the table [24] and Fig we can infer that 0.0015 is the best learning rate for Cat-Resnet18 combination.

Future scope of this project would be to replicate the results with the new learning rate for all 8 other instances and take a shot at continual learning for studying the tradeoffs between scratch training and transfer learning.

References

[1] "Howard, andrew au - zhu, menglong au - chen, bo au - kalenichenko, dmitry au - wang, weijun au - weyand, tobias au - andreetto, marco au - adam, hartwig py 2017/04/16 sp - t1 mobilenets: Efficient convolutional neural networks for mobile vision applications er," 1

[2] "He, kaiming au - zhang, xiangyu au - ren, shaoqing au - sun, jian py 2016/06/01 sp - 770 ep - 778 t1 - deep residual learning for image recognition do - 10.1109/cvpr.2016.90 er," 1

[3] "Vasudevan, varun au - bassenne, maxime au - islam, md tauhidul au - xing, lei py - 2022/01/29 sp - t1 - image classification using graph neural network and multiscale wavelet superpixels er," 1

[4] "Ibraheam, mai & gebali, fayez & li, kin fun & sielecki, leonard. (2020). animal species recognition using deep learning. 10.1007/978-3-030-44041-1$_47.c,''$ 1

[5] "Singh, dalwinder & singh, birmohan. (2019). investigating the impact of data normalization on classification performance. applied soft computing. 105524. 10.1016/j.asoc.2019.105524.," 1, 4

[6] "Yang, li shami, abdallah. (2020). on hyperparameter optimization of machine learning algorithms: Theory and practice.," 1

[7] "Agarwal, ambuj & kiran, vidhu & jindal, rupesh & chaudhary, deepak tiwari, raj. (2022). international journal of intelligent systems and applications in engineering optimized transfer learning for dog breed classification.," 2

[8] "H. nguyen et al., "animal recognition and identification with deep convolutional neural networks for automated wildlife monitoring,"," 2017 ieee international conference on data science and advanced analytics (dsaa), 2017, pp. 40-49, doi: 10.1109/dsaa.2017.31.," 2

[9] "Yang, l., song, s. and chen, c.p., 2018, october. transductive transfer learning based on broad learning system. in 2018 ieee international conference on systems, man, and cybernetics (smc) (pp. 912-917). ieee," 2

[10] "Liu, j., kanazawa, a., jacobs, d., belhumeur, p. (2012). dog breed classification using part localization. in: Fitzgibbon, a., lazebnik, s., perona, p., sato, y., schmid, c. (eds) computer vision – eccv 2012. eccv 2012. lecture notes in computer science, vol 7572. springer, berlin,heidelberg," 2

[11] "Chen, guobin et al. "deep convolutional neural network based species recognition for wild animal monitoring." 2014 ieee international conference on image processing (icip) (2014): 858-862.," 2

[12] "Yue, zhenjia & ma, liangping & zhang, runfeng. (2020). comparison and validation of deep learning models for the diagnosis of pneumonia. computational intelligence and neuroscience. 2020. 1-8. 10.1155/2020/8876798.," 2

[13] 2 http://vision.stanford.edu/aditya86/ImageNetDogs/.

[14] "O. ulucan, d. karakaya and m. turkan, "a large-scale dataset for fish segmentation and classification," 2020 innovations in intelligent systems and applications conference (asyu), 2020, pp. 1-5, doi: 10.1109/asyu50717.2020.9259867.," 2

[15] 2 https://www.kaggle.com/competitions/dog-breed-identification/data.

[16] 2 https://www.kaggle.com/datasets/ma7555/cat-breeds-dataset.

[17] 2 https://www.kaggle.com/datasets/crowww/a-large-scale-fish-dataset.

[18] "D. patel et al., "flops: On learning important time series features for real-valued prediction," 2020 ieee international conference on big data (big data), 2020, pp. 1624-1633, doi: 10.1109/bigdata50022.2020.9378499.," 3

[19] "Kingma, diederik ba, jimmy. (2014). adam: A method for stochastic optimization. international conference on learning representations.," 3

[20] 3 https://towardsdatascience.com/parameters-and-hyperparameters-aa609601a9ac.

[21] 6 https://mlcheatsheet.readthedocs.io/en/latest/loss_functions.html.

[22] 6 https://towardsdatascience.com/estimating-optimal-learning-rate-for-a-deep-neural-network-ce32f2556ce0.

[23] 6 https://www.cs.toronto.edu/~hinton/absps/tsne.pdf.

[24] "van der maaten, laurens hinton, geoffrey. (2008). vivualizing data using t-sne. journal of machine learning research. 9. 2579-2605.," 6

A. Appendix

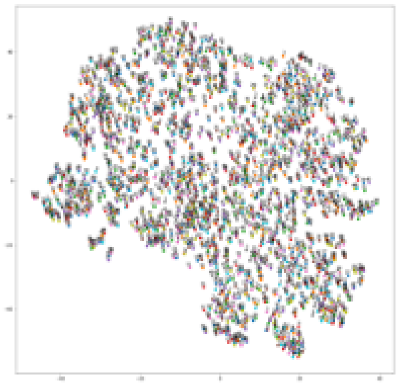
Figure 11. Scatter plot of TSNE features of Dog-ResNet18.

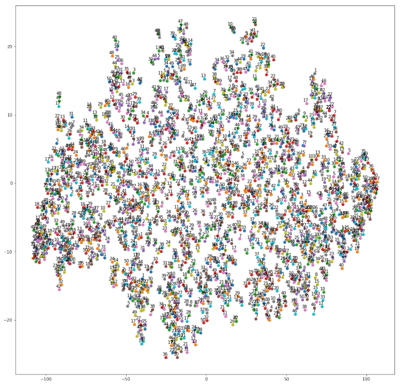
Figure 12. Scatter plot of TSNE features of Dog-ResNet18 (TL).

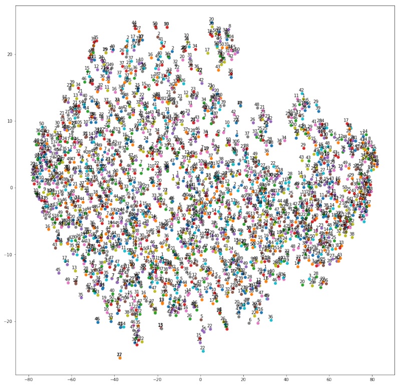
Figure 13. Scatter plot of TSNE features of Dog-MobilenetV2

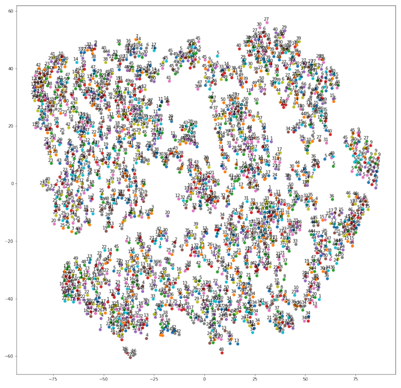
Figure 14. Scatter plot of TSNE features of Dog-MobilenetV2 (TL).