DD2424 Project report Draft

Bashar Levin, Johannes Joujo, Nasir Alizade

May 2025

1 Abstract

This project explores the use of transfer learnign for image classification using the Oxford-III pet dataset. By finetuning a pretrained ResBet18 model the project aims to solve both a classification problem between cats and dogs and a more advanced multi class problem that deals with 37 different breeds. The study test various finetuning methods, such as layer freezing and even test the impact of different optimizers such as Adam and NGA. Both experiments were successful and showed high accuracy. For the multiclass experiment unfreezing two blocks and using Adam optimizer gave the best results.

2 Introduction

This project is about transfer learning. Transfer learning is one of the most popular uses of deep learning. Transfer learning has many benefits, ranging from addressing environmental issues to further improving a model's capabilities.
By using the pretrained convolutional neural network ResNet18 with Oxford-III Pet dataset our goal is to

- Binary classification: distinguishing between cats and dogs.
- Multi-class classification: identifying 37 individual pet breeds.

This project is important because we get to learn how to fine-tune a model that generates high accuracy for both simple and harder problems.

3 Related Work

In the article *DTL-I-ResNet18: facial emotion recognition based on deep transfer learning and improved ResNet18* by Rabie et al. [1] showed that they were able to get the best result on facial emotion recognition with the help of transfer learning from the ResNet18 model while also testing other deep CNN models such as DenseNet121, MobileNet, ResNet101, ResNet50 and SENet18.

Another similar work conducted by Miladinovic et al. in the article *Evaluating deep learning models for classifying OCT images with limited data and noisy labels* [3] showed that all their tested models generated a test accuracy of more than 90% wen trained on a data set containing 345 or more images. The evaluated deep learning (DL) architectures were ResNet18, ResNet34, ResNet50, VGG16, and InceptionV3. Out of the tested DL models InceptionV3 was the best performing when trained on all the available training data.

4 Data

The data used to fine-tune the ResNet18 pre-trained model is the Oxford-IIIT Pet Dataset. The dataset consists of 37 categories, each having 200 samples per class. The images are different in size, pose and lightning. The images have annotation of breed, a tight bounding box region of interest, a pixel level foreground-background segmentation (Trimap).

The dataset have a total number of 7349 images of cats and dogs. 4978 out of the images are of dogs and the remaining 2371 images are of cats. There are 25 dog breeds and 12 cat breeds.
On the website *paperswithcode* [4] some of the state of the art models have been ranked on how good accuracy they receive on the dataset. The best model is OmniVec2 followed by OmniVec with an accuracy of 99.6% and 99.2%. The worst performing state of the art model is SEER (RegNet10B) with an accuracy of 85.3%.

4.0.1 Preprocess

The preprocess was done to prepare the dataset for binary and multi-class classification tasks.

4.0.2 Binary preprocessing

The dataset provided a file (list.txt), which contains metadata about images, including species (1 for cats, 2 for dogs). Species values are mapped to binary labels, with 0 assigned to cats and 1 to dogs. Each image file is validated to ensure it existed in the images directory. The dataset is subsequently split into training(70%), validation(15%) and test(15%) sets. Finally, images are copied into cat or dog subdirectories under train,val and test folders.

4.0.3 Multi-class preprocessing

For multi-class classification, the same file used here as in binary preprocesing, but extracted images names to extract breed names, such as *yorshire_terrier*. The dataset is then divided into training,validation and test sets using stratified sampling based on the breeds labels. Images are copied into breed-specific subdirectories under training, val, and test folders.

5 Method

In this section a brief explanation of the experiment set up is provided.

5.1 Binary

The binary classification involves data loading, model setup, training, and evaluation, implemented using Pytorch and torchvision.
The dataset, stored in a folder and already been preprocessed. Data loaders are created using a utility function, *get_dataloaders*, which applies standard image tranfomation and returns training and validation data loaders with batch size of 32. The datasets consist of two classes cat and dog.

A ResNet18 model, pretrained on ImageNet, is used ad the baseline architecture. The fully connected layer is modified to output two classes, corresponding to binary classification tasks. All layers except the final fully connected layer are frozen to leverage pretrained weights, and only the parameters are set to require gradient updates. The Adam optimizer is configured with a learning rate of 0.001, and the cross-entropy loss function is used for training.

5.2 Multiclass

In this experiment, the goal was to identify the correct bread out of 37 possible categories in the Oxford-III dataset. In order to do this, finetuning a pretrained ResNet18 model was implemented. Firstly, the original FC layer of ResNet18 which was trained for 1000 classes on ImageNet [2] was replaced with a linear layer that corresponds to 37 pet breed classes. This made it possible for the model to generate breed prediction as output that is specific to our database. Furthermore multiple experiments were run and evaluated.

6 Experiment

In this section the details of all experiments are displayed.

6.1 Multiclass

Multiple experiments of layer freezing were run to evaluate different finetuning strategies.

- **Final layer**All pretrained layers were frozen except for the final new added linear layer.
- **Last two layers (4, 5)**Unfreeze the last one or two layers while keeping the rest frozen.
- **Unfreeze all**All layers here were not frozen making them fully trainable.

Training was done using cross entropy loss function and two optimizers, the optimizers used in this case are (Adam) and (NAG). Gradient accumulation was also used to simulate smaller batch size. Each experiment were run for a fixed number of epoches (20). Early stopping is also implemented to avoid overfitting. Lastly TensorBoard was used to log the training matrices and to plot the accuracy curves.
Model setup initially: epoch=15, learning rate new layer= 1e-4,learning rate pretrained layer=1e-3 and batch size of 32. Se Table 1, showing the best validation accuracy.

Experiment	Best validation	Model setup
1	0.8854	resnet18_baseline_fc_only
2	0.9161	resnet18_unfreeze_1block_adam
3	0.9251	resnet18_unfreeze_2blocks_adam
4	0.9305	resnet18_unfreeze_all_adam
5	0.8917	resnet18_unfreeze_2blocks_nag height

Table 1: Experiment, best validation and model setup

6.2 Binary

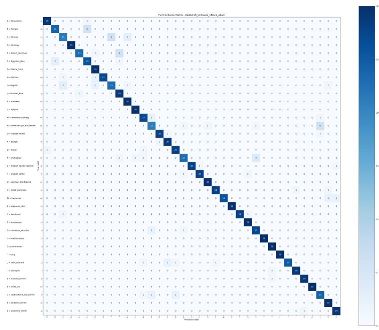
For the binary problem, all layers in the pre-trained ResNet model were frozen exepct the final FC layer it was instead replaced with a new linear layer that outputs two classes. The data was proccessed into binary labels (0 for cats and 1 for dogs) and was even split into (70 traning, 15% validation and 15 test). And they were saved using PyTorch dataLoader. The traning was done for 30 epoches and with a 0.0001 learning rate using cross entropy function and adam optimizer. Only the parameters in the last layer were changed because the rest of the layers were frozen during training.

7 Result

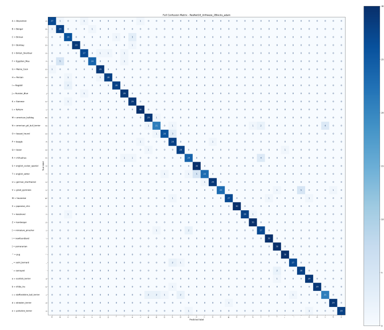
The binary classification experiment achieved a test accuracy of 0.9918 using a ResNet18 model with all layers frozen except the final fully connected layer, trained with the Adam optimizer. For the multi-class classification, five fine-tuned strategies were evaluated, with results summarized in Table 2. The best performance was obtained by unfreezing two blocks of ResNet18 and using the Adam optimizer, yielding a test accuracy of 0.9288.

Figure 1 illustrates the models' performance across the 37 breeds, with the two-block unfreezing strategy, Figure 1b, showing the most balanced predictions. Training history plots, Figure 2, reveal stable convergence for all models, with the Adam-based strategies generally outperforming NAG in terms of validation accuracy and loss.

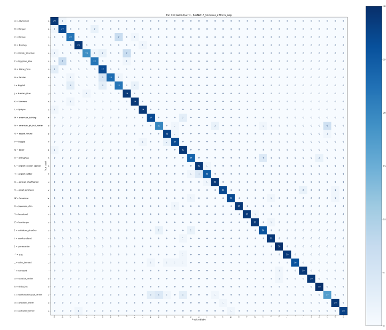
Figure 2 illustrates the training, validation loss, and accuracy for each strategy. The two-block unfreezing strategy with Adam, Figure 2b, shows the most balanced and stable performance, with loss curves steadily decreasing and accuracy showing a clear upward trend. The binary baseline, Figure 2e, showed stable accuracy.



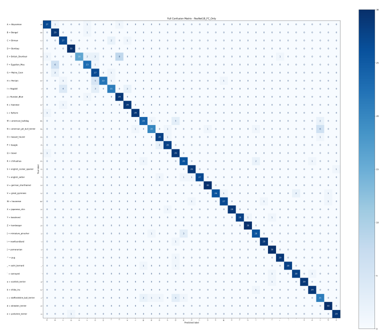
(a) ResNet18 unfreeze 1 block (Adam)



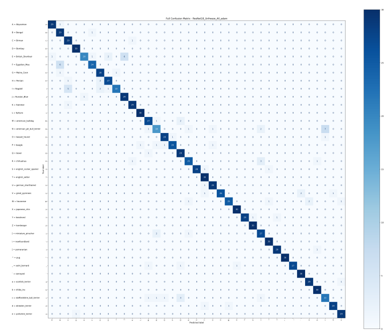
(b) ResNet18 unfreeze 2 blocks (Adam)



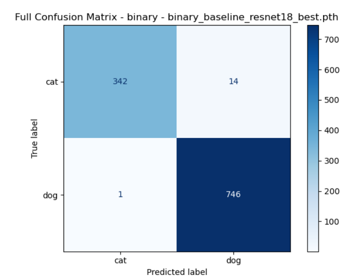
(c) ResNet18 unfreeze 2 blocks (NAG)



(d) ResNet18 FC only

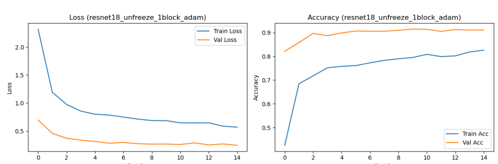


(e) ResNet18 unfreeze all (Adam)

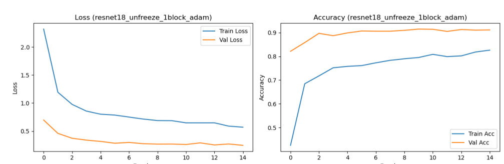


(f) ResNet18 binary baseline

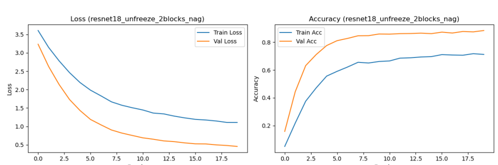
Figure 1: Confusion matrices for different fine-tuning strategies.



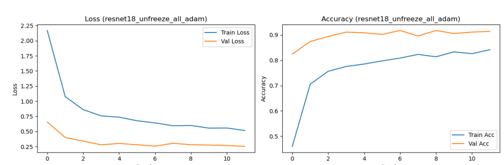
(a) ResNet18 unfreeze 1 block (Adam)



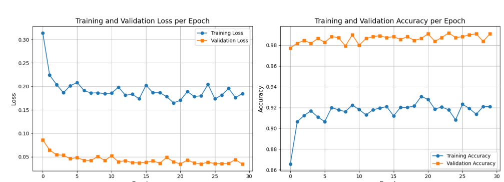
(b) ResNet18 unfreeze 2 blocks (Adam)



(c) ResNet18 unfreeze 2 blocks (NAG)



(d) ResNet18 unfreeze all layers (Adam)



(e) ResNet18 binary baseline training

Figure 2: Training and validation loss/accuracy for different fine-tuning strategies.

Strategies	Test accuracy	Optimizer
ResNet18 Unfreeze 2 Blocks	0.9288	Adam
ResNet18 Unfreeze 1 Block	0.9206	Adam
ResNet18 FC Only	0.8972	Adam
ResNet18 Unfreeze All	0.9261	Adam
ResNet18 Unfreeze 2 blocks	0.8972	nag
Binary freeze all except last block	0.9918	Adam

Table 2: Test accuracy

8 Conclusion

The experiments were successful and gave satisfactory results. When it comes to multi-class classification the results show that the best approach is to unfreeze 2 blocks and train with Adam optimizer, this set up gave test accuracy as high as 0.9288. Further more the binary classification gave also satisfactory results when training with the same setup, the test results for binary was 0.9918.

References

[1] Wen He, Yuming Zhang, Pengfei Zhang, Xin Zhao, and Xue Li. Msfe-net: a multi-scale feature extraction network for ship classification in sar images. *Journal of Electronic Imaging*, 32(6):061408, 2023.

[2] Shaoqing Ren Jian Sun Kaiming He, Xiangyu Zhang. Deep residual learning for image recognition. `https://arxiv.org/pdf/1512.03385`, 2015.

[3] Ziyuan Li, Yaxin Li, Yang Liu, Wei Wang, Jing Zhang, and Jian Zhang. Deep learning for automated classification of optical coherence tomography images in ophthalmology: A review. *Diagnostics*, 14(10):1019, 2024.

[4] Omkar M. Parkhi, Andrea Vedaldi, Andrew Zisserman, and C. V. Jawahar. Cats and dogs. `https://www.robots.ox.ac.uk/~vgg/data/pets/`, 2012.