DD2424 Deep Learning in Data Science
Transfer Learning

Ksenia Biazruchanka
20050523-T068
kseniabi@ug.kth.se

Tobias Haunreiter
19990923-T574
thau@ug.kth.se

Alexander Soukup
20020103-T333
ansoukup@ug.kth.se

Johannes Jeup
20010303-T498
jeup@ug.kth.se

Abstract

This project explores transfer learning techniques for image classification using the Oxford-IIIT Pet Dataset. A pre-trained ResNet50 model is fine-tuned to perform both binary (cat vs. dog) and multi-class breed classification. Additionally, the FixMatch semi-supervised learning algorithm is evaluated to enhance performance with limited labeled data.

1 Introduction

This project is aimed to fine-tune a pre-trained Convolutional Neural Network(CNN) (e.g. ResNet50), for a new image classification task. The target data set is the Oxford-IIIT Pet Dataset [8], which contains approximately 200 images in each of its 37 categories. The initial goal is to develop a binary classifier capable of distinguishing between cats and dogs and to extend the network to predict all pet breeds.

For the classification task (binary + multi-class) different strategies are explored to boost performance in terms of accuracy. These include simultaneously fine-tuning the final layers of the network and applying gradual un-freezing. There will also be a study on network architecture robustness using imbalanced classes. Furthermore, we will present an investigation into a SSL approach using the FixMatch algorithm.

2 Related Work

Convolutional Neural Networks (CNNs) have become the cornerstone of image classification tasks due to their ability to automatically learn hierarchical feature representations from raw image data. Notable among these are GoogLeNet (Inception Module) [13], Residual Network (ResNet) [4], DenseNet [5], and VGGNet (Visual Geometry Group Network) [10]. ResNet [4] introduces residual connections that enable the training of very deep networks by mitigating vanishing gradients. Its skip connections allow information to bypass layers, improving optimization and performance. ResNet variants, like ResNet-50, are widely used as backbones in image classification due to their robustness and scalability.

SSL aims to leverage a small amount of labeled data alongside a larger quantity of unlabeled data to improve model performance. Several SSL methods have been developed with varying complexity and performance. Early approaches such as the Π-Model [9] and Pseudo-Labeling [7] use consistency regularization and self-training mechanisms, respectively. Mean Teacher [14] improves upon these by using an exponential moving average of model parameters. More recent and sophisticated methods like MixMatch [3], UDA (Unsupervised Data Augmentation) [16], and ReMixMatch [2] integrate stronger augmentation strategies and additional training objectives.

The FixMatch algorithm [11] simplifies the SSL pipeline by combining pseudo-labeling with consistency regularization. It generates high-confidence pseudo-labels on weakly augmented images and trains the model to match these labels on strongly augmented versions. Despite its simplicity, FixMatch achieves state-of-the-art results on several benchmark datasets. Table 1 summarizes the error rates reported for various methods.

Method	CIFAR-10 (250)	CIFAR-100 (2500)	SVHN (1000)	STL-10 (1000)
Π-Model	54.26%	57.25%	7.54%	26.23%
Pseudo-Labeling	49.78%	57.38%	9.94%	27.99%
Mean Teacher	32.32%	53.91%	3.42%	21.43%
MixMatch	11.05%	39.94%	3.50%	10.41%
UDA	8.82%	33.13%	2.46%	7.66%
ReMixMatch	**5.44%**	**27.43%**	**2.65%**	**5.23%**
FixMatch (RA)	**5.07%**	28.29%	**2.28%**	7.98%

Table 1: Test error rates on standard SSL benchmarks using 250–2500 labeled examples (adapted from [11]). The reported accuracy is the mean accuracy of five training runs.

FixMatch stands out due to its conceptual and implementation simplicity. Unlike ReMixMatch or UDA, it requires fewer hyperparameters and does not rely on label sharpening, distribution alignment, or training signal annealing. Despite these simplifications, FixMatch delivers comparable or superior results, particularly on CIFAR-10 and SVHN. This balance of simplicity and effectiveness makes FixMatch an ideal choice for applying SSL to new domains. Consequently, this project explores the performance of FixMatch on the Oxford-IIIT Pet dataset [8], extending its evaluation to a domain with different visual features and class characteristics than the commonly-used benchmarks.

3 Data

The dataset used in this project is the Oxford-IIIT Pet dataset [8], which is designed for fine-grained object categorization. It is a public dataset with 37 categories of pet breeds (12 cats, 25 dogs) and a total of 7349 pet images. Each class includes around 200 images. For each image, the dataset provides three types of annotations: a breed label, a pixel-level trimap segmentation (foreground, background, and ambiguous), and a bounding box marking the head region. A small sample size of the images is shown in figure 1. The dataset is partitioned into training, validation, and test sets.

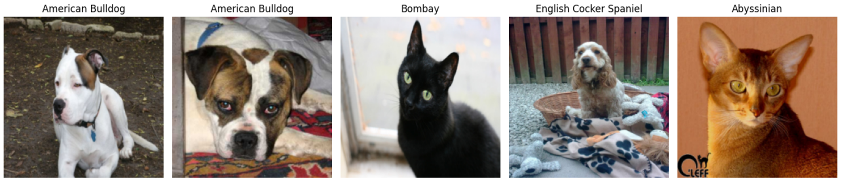

Figure 1: Random Samples from the Oxford-IIIT Pet Dataset. The breed name is written above each of the images.

In the Table 2 we present state-of-the-art performance on the Oxford-IIT Pet dataset with different algorithms Supervised Learning (SL), Semi-Supervised Learning (SSL) and Transfer Learning(TL) utilizing ResNet and other models.

Algorithm	Paper	Model	Extra Training Data	Accuracy (%)
SSL, TL	Srivastava, Sharma (2024) [12]	OmniVec2	Yes	99.60%
SL, TL	Qin Xu et al. (2023) [17]	IELT	No	95.28%
SL, TL	Kolesnikov et al. (2019) [6]	BiT-L (ResNet)	No	96.62 %
SSL	Feng Wang et al. (2021) [15]	TWIST (ResNet50)	No	94.5%

Table 2: State-of-the-art performance on Oxford-IIT PET dataset. Results are adapted from Papers with Code benchmark.

The primary motivation behind the dataset is to address the challenge of fine-grained classification, where differences between classes (i.e., pet breeds) are subtle and intra-class variability is high. The Oxford-IIIT Pet dataset introduces a new level of complexity in SSL due to its fine-grained labels and high intra-class variance, so it was chosen to evaluate the performance of FixMatch on a domian that is different from the commonly used SSL benchmarks such as CIFAR-10/100. Testing FixMatch in this setting enables assessment of the algorithm's ability to generalize to more challenging, real-world classification problems where labeled data may be scarce and class boundaries are less distinct.

4 Methods

4.1 Binary Classification using Feature Extraction

To solve the binary classification problem of recognizing pictures of Dog vs Cat, we adopted the transfer learning approach by using a pre-trained convolution network. The main strategy was to freeze all convolution layers and train only the final fully connected layer, also known as Feature Extraction. The model was optimized using Adaptive Moment Estimation (Adam) optimizer. The training was performed using mini-batch gradient decent and Binary Cross-Entropy (BCE) loss. To monitor generalization, we evaluated the model on a validation set after each epoch, computing both the loss and accuracy. This helped ensure the model did not overfit.

4.2 Breed Classification

To solve the multi-class classification problem of recognizing 37 breed of cat and dog, the final fully connected layer was modified accordingly. To cope with the increased difficulty of this classification task two main strategies to fine tune the network were evaluated.

The first strategy aimed at training l of the last layers (+ the classification layer) simultaneously instead of training only the last layer as done for the binary classification task. Therefore several experiments have been carried out with increasing values for l until no significantly improvements could be observed.
The second strategy implemented gradual un-freezing. Therefore one layer of the network was unfrozen for multiple epochs at a time during training, starting from the last layer and progressively unfreezing earlier layers. Layers without trainable parameters, e.g. pooling layers, were skipped. For both strategies the Adam optimizer was utilized.
To further enhance performance several potential improvements were considered, such as different learning rate scheduling for different layers, applying data augmentation and L2 regularization during training as well as the effect of fine-tuning or not the batch-norm layers.

In order to analyze the robustness of the trained network and for further fine-tuning imbalanced classes were considered. Therefore, we reduced the number of images for each cat breed by 80%. To compensate for the imbalanced training set we applied both weighted cross-entropy and over-sampling of the minority classes.

More information on these methods is presented in A.2.

4.3 SSL: The FixMatch Method

The FixMatch method was introduced by Sohn et al. [11]. The aim with this method is to use only a few labeled samples per class and have a lot of unlabeled samples, while still being able to get high accuracy results for the classification task. The approach for FixMatch is to combine consistency regularization and pseudo-labeling and exploiting weak and strong augmentation when performing consistency regularization. The fine details are described in [11], but a brief overview on the methodology is given here.

Consistency regularization - first proposed in [1] - leverages unlabeled data by assuming similar network outputs for augmented versions of the sample, to enhances robustness and generalization. In parallel, pseudo-labeling uses the model's own predictions as artificial labels for unlabeled data. Only predictions with high confidence (above a threshold) are retained, and a cross-entropy loss is applied using the most probable class (i.e., a "hard" label). This method encourages the model to produce confident, low-entropy predictions, reinforcing learning from unlabeled examples.

Loss Function The FixMatch algorithm has two forms of cross-entropy loss. A supervised loss l_S applied to labeled data and an unsupervised loss l_u for the unlabeled data samples. The supervised loss is the standard cross-entropy loss on weakly augmented samples

$$l_s = \frac{1}{B}\sum_{b=1}^{B} H(p_b, p_m(y|\alpha(x_b))). \tag{1}$$

The artificial labels for each unlabeled example are obtained by using the network's predicted class distribution given a weakly-augmented unlabeled image $q_b = p_m(\alpha(u_b))$. Afterwards, $\hat{q}_b = \texttt{arg max}(q_b)$ is used to get the pseudo-label, except for strongly augmented versions of u_b

$$l_u = \frac{1}{\mu B}\sum_{b=1}^{\mu B} \underline{\mathbf{1}}(\max(q_b) \geq \tau) H(\hat{q}_b, p_m(y|A(u_b))), \tag{2}$$

where τ is a scalar hyperparameter denoting the threshold for which the pseudo-label is retained. The overall loss to be minimized is

$$l = l_S + \lambda_u l_u, \tag{3}$$

where λ_u is a fixed scalar hyperparameter, representing the relative weight of the unlabeled loss compared to the labeled loss. Equation 2 introduces a form of consistency regularization contributing to FixMatch's success.

Augmentation FixMatch takes advantage of two different augmentation techniques - weak and strong augmentation. The definition of weak augmentation in the context of this project is, besides the normalization for all images, random horizontal flipping and a random resize-crop to fit the correct dimensionality of $224 \times 224 \times 3$ per image. The strong augmentation involves weak augmentation and methods as random color jitter ($p = 0.8$), gaussian blur ($p = 0.5$) and random rotation ($p = 0.5$) of up to $\pm 15\,\mathrm{deg}$. Ten randomly selected images are shown in 2 with the raw image being in the first row, weak augmentation is applied to the second row and strong augmentation to the third row. It is important to note, that all images are "de-normalized" again, as it is easier to see the difference for humans. However, the network "sees" the normalized images during training.

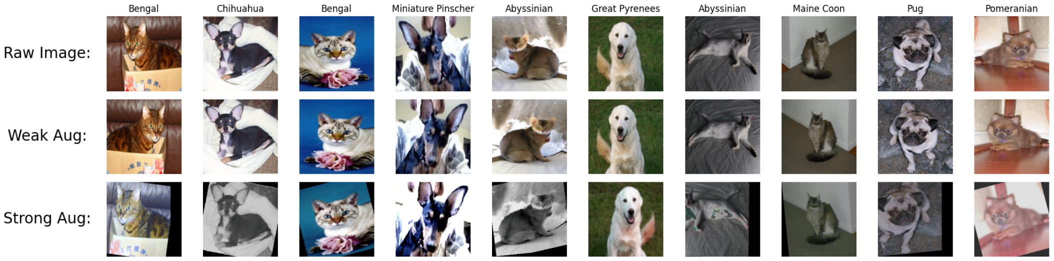

Figure 2: Weak and strong augmentation visualized. In the first row are the raw images displayed, to those the augmentation methods weak (second row) and strong (third) row are applied. All images are, for better human interpretability, de-normalized.

5 Experiments

5.1 Binary Classification

To achieve high accuracy in the binary classification task of distinguishing between cats and dogs, ResNet50 model was used. It was optimized using Adam optimizer with learning rate $\eta = 10^{-4}$. The best achieved validation accuracy was **99.45%**. The best model was tested on the test dataset and was able to achieve a performance of **99.29%**. The learning curves (Training and Validation Loss and Accuracy) are presented in A.1.

5.2 Breed Classification

The best validation accuracy achieved for the multiclass classification when only training the last layer for 25 epochs is **90.76%**. The result was achieved using ResNet50, the following hyperparameters: $\lambda = 0.001$, $\eta = 10^{-4}$, and data augmentation. The model achieved a test accuracy of **89.15%**.

5.2.1 Fine-tuning L layers simultaneously

ResNet50 was chosen as the base model for all experiments. The following hyperparameters were used: $\lambda = 0.001$, $\eta = 10^{-4}$. Data augmentation was applied to improve generalization. Due to the high risk of overfitting during fine-tuning, each experiment was limited to a small number of epochs (`num_epochs = 5`). The results are summarized in Table 3.

L	Train Accuracy(%)	Val Accuracy (%)
0	86.68%	84.38%
2	97.76%	94.70%
3	98.27%	94.43%
4	97.69%	94.43%
5	98.40%	94.29%
8	98.03%	92.66%
9	97.93%	93.61%

Table 3: Fine-tuning L layers simultaneously. With Data Augmentation and L2 generalization.

As shown in the results, fine-tuning a larger number of layers tends to reduce validation accuracy, suggesting overfitting when too many layers are trained at once. The best model achieved a test accuracy of **91.33%**.

5.2.2 Gradual un-freezing

The same experimental setup was used as in the previous section. In this strategy, layers were unfrozen incrementally, and each newly unfrozen layer was trained for 5 epochs while keeping the rest fixed. The best training and validation accuracies after unfreezing each layer are reported in Table 4.

L	Train Accuracy(%)	Val Accuracy (%)
0	87.40%	84.10%
2	97.83%	93.75%
3	99.46%	95.92%
4	99.80%	96.06%
5	99.90%	96.06%
8	99.99%	96.06%
9	99.97%	95.92%

Table 4: Gradually un-freezing L-th layer. With Data Augmentation and L2 generalization.

Compared to simultaneous fine-tuning, the gradual un-freezing strategy yielded better validation performance. The results suggest that fine-tuning up to $L = 4$ achieves optimal results, beyond which overfitting may occur. The best model (fine-tunning up to $L = 4$) achieved a test accuracy of **91.91%**.

5.2.3 Further Improvements

To enhance network performance, several strategies were evaluated, including data augmentation, learning rate scheduling, and batch normalization fine-tuning. For all experiments ResNet50 was used with gradual un-freezing up to $L = 3$, training each layer for 3 epochs. Table 5 summarizes the results.

Run	Data Aug.	LR Strategy	Batch Norm	Train. Acc.	Val. Acc.
1	False	const	False	99.59%	94.16%
2	False	const	True	99.69%	94.29%
3	True	const	True	98.85%	94.84%
4	True	layer_decr	True	98.34%	94.97%
5	True	layer_epoch_decr	True	97.21%	94.16%

Table 5: Improvements of network performance.

Enabling data augmentation and batch norm fine-tuning improves validation accuracy, while learning rate schedules help reduce overfitting, though overly aggressive decay may hinder training. The effect of L2 regularization can be found in A.2 in Table 7. The best final test accuracy computed to **92.37%** and was achieved using gradual un-freezing of $L = 4$ layers for 3 epochs per layer with data augmentation, L2 regularization with $\lambda = 0.01$ and an exponentially decaying learning rate per layer. Further results and more insight into the additions and their effect is found in A.2, as well as the learning curves for the best settings in Figure 9.

5.2.4 Imbalanced Classes

To enhance comparability, the same parameter settings were used as in the best run using fine-tuning in section A.2. Firstly, imbalance classes were applied without any compensation strategy. The achieved test accuracy was 86.15% with fine-tuning up to layer 5. Considering compensation, the achieved test accuracy is 89.64% with the weighted cross-entropy loss approach and **89.70%** with the the weighted sampling strategy. Therefore, both methods prove to be equally effective in dealing with the imbalanced data set, although neither is able to compensate completely. The corresponding learning curves are found in A.2.

5.3 SSL: FixMatch

We defined the threshold for which the pseudo-label is retained to $\tau = 0.95$ and chose $\lambda_u = 1$.

Due to limited computational resources, we applied a simultaneous fine-tuning strategy. The algorithm was trained for 10 epochs, except for the cases with 20 and 50 images per class, which were limited to 5 epochs due to computational constraints. Table 6 presents test accuracies with varying amounts of labeled data. The “Images per Class” column indicates how many labeled samples were available per class, while “Labeled Data (%)” expresses this as a percentage of the total training dataset. “Supervised Only” shows the accuracy of a model trained using only the labeled subset and the ’basic’ technique, and “FixMatch” reports performance after applying the FixMatch algorithm.

Images per Class	Labeled Data (%)	Supervised Only	FixMatch
50	60%	89.48%	90.35%
20	25%	88.42%	87.64%
10	12.5%	70.24%	86.41%
2	2.5%	11.99%	72.91%
1	1%	5.56%	50.15%

Table 6: Test accuracy with limited labeled data. Comparison of ’basic’ technique and FixMatch.

FixMatch significantly improves performance in low-label regimes (1–15%), where the baseline model performs poorly. The FixMatch algorithm was run for only 10 epochs, slightly higher accuracy can likely be achieved with longer training. Compared to the results by [11], performance on this dataset remains lower overall, which can be attributed to the increased difficulty of the Oxford-IIIT Pet dataset—marked by fine-grained classes and high intra-class variability. The Loss and Accuracy training curves for 2 Images per Class experiment is shown in Figure 3. Similar plots for other experiments are found in A.3.1. We report both top-1 and top-5 validation accuracies, although top-5 is less informative in this context due to the small number of classes.

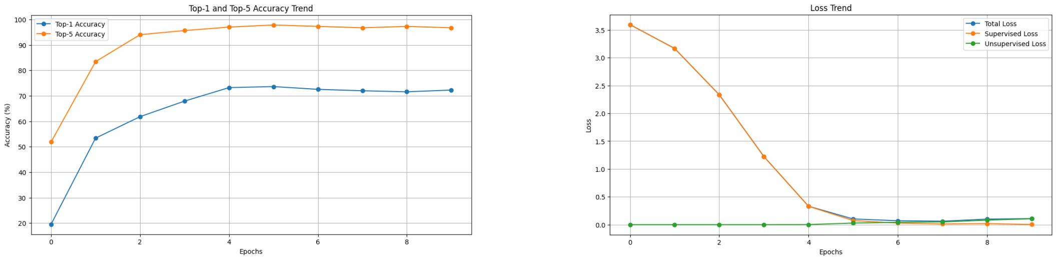

Figure 3: Accuracy and Loss Training curves for FixMatch 2 labeled images per class.

6 Conclusion

This project demonstrated the effectiveness of transfer learning and semi-supervised learning for image classification on the Oxford-IIIT Pet Dataset. Fine-tuning a pre-trained ResNet50 model achieved high performance in both binary (cat vs. dog) **>99%** and multi-class breed classification **>92%** tasks. Applying the FixMatch algorithm showed promising results under conditions of limited labeled samples, achieving a accuracy of **>72%** using only 2 labeled images per class.

References

[1] Philip Bachman, Ouais Alsharif, and Doina Precup. Learning with pseudo-ensembles. *Advances in neural information processing systems*, 27, 2014.

[2] D Berthelot, N Carlini, ED Cubuk, A Kurakin, K Sohn, H Zhang, and C Raffel. Remixmatch: Semi-supervised learning with distribution matching and augmentation anchoring iclr. *OpenReview. net*, 2020.

[3] David Berthelot, Nicholas Carlini, Ian Goodfellow, Nicolas Papernot, Avital Oliver, and Colin A Raffel. Mixmatch: A holistic approach to semi-supervised learning. *Advances in neural information processing systems*, 32, 2019.

[4] 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.

[5] Gao Huang, Zhuang Liu, Laurens Van Der Maaten, and Kilian Q Weinberger. Densely connected convolutional networks. In *Proceedings of the IEEE conference on computer vision and pattern recognition*, pages 4700–4708, 2017.

[6] Alexander Kolesnikov, Lucas Beyer, Xiaohua Zhai, Joan Puigcerver, Jessica Yung, Sylvain Gelly, and Neil Houlsby. Big transfer (bit): General visual representation learning. In *Computer Vision–ECCV 2020: 16th European Conference, Glasgow, UK, August 23–28, 2020, Proceedings, Part V 16*, pages 491–507. Springer, 2020.

[7] Dong-Hyun Lee et al. Pseudo-label: The simple and efficient semi-supervised learning method for deep neural networks. In *Workshop on challenges in representation learning, ICML*, volume 3, page 896. Atlanta, 2013.

[8] Omkar M Parkhi, Andrea Vedaldi, Andrew Zisserman, and CV Jawahar. Cats and dogs. In *2012 IEEE conference on computer vision and pattern recognition*, pages 3498–3505. IEEE, 2012.

[9] Antti Rasmus, Mathias Berglund, Mikko Honkala, Harri Valpola, and Tapani Raiko. Semi-supervised learning with ladder networks. *Advances in neural information processing systems*, 28, 2015.

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

[11] Kihyuk Sohn, David Berthelot, Nicholas Carlini, Zizhao Zhang, Han Zhang, Colin A Raffel, Ekin Dogus Cubuk, Alexey Kurakin, and Chun-Liang Li. Fixmatch: Simplifying semi-supervised learning with consistency and confidence. *Advances in neural information processing systems*, 33:596–608, 2020.

[12] Siddharth Srivastava and Gaurav Sharma. Omnivec2-a novel transformer based network for large scale multimodal and multitask learning. In *Proceedings of the IEEE/CVF conference on computer vision and pattern recognition*, pages 27412–27424, 2024.

[13] Christian Szegedy, Wei Liu, Yangqing Jia, Pierre Sermanet, Scott Reed, Dragomir Anguelov, Dumitru Erhan, Vincent Vanhoucke, and Andrew Rabinovich. Going deeper with convolutions. In *Proceedings of the IEEE conference on computer vision and pattern recognition*, pages 1–9, 2015.

[14] Antti Tarvainen and Harri Valpola. Mean teachers are better role models: Weight-averaged consistency targets improve semi-supervised deep learning results. *Advances in neural information processing systems*, 30, 2017.

[15] Feng Wang, Tao Kong, Rufeng Zhang, Huaping Liu, and Hang Li. Self-supervised learning by estimating twin class distribution. *IEEE Transactions on Image Processing*, 32:2228–2236, 2023.

[16] Qizhe Xie, Zihang Dai, Eduard Hovy, Thang Luong, and Quoc Le. Unsupervised data augmentation for consistency training. *Advances in neural information processing systems*, 33:6256–6268, 2020.

[17] Qin Xu, Jiahui Wang, Bo Jiang, and Bin Luo. Fine-grained visual classification via internal ensemble learning transformer. *IEEE Transactions on Multimedia*, 25:9015–9028, 2023.

A Additional Results

A.1 Binary Classification

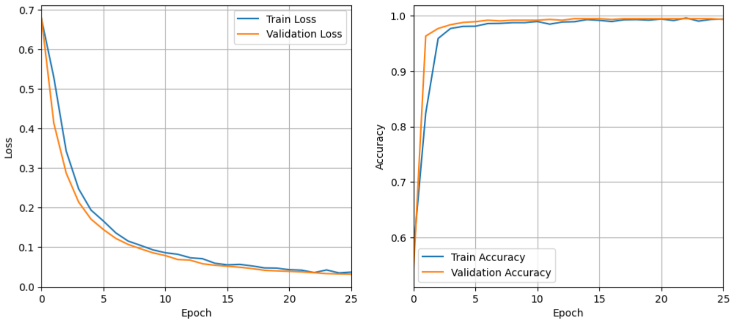

Figure 4: Training Curves (Loss, Accuracy) for Binary Classification Experiment using ResNet50.

A.2 Breed Classification

The following section gives further insight into our findings regarding the breed classification and analyzes the effect of the additional improvements applied to the multi-class network. Figure 5 and 6 give a first impression on the behavior of the adapted ResNet18 for a long training run. Both strategies suggest a strong overfitting on the training set and produced comparable results as already seen in 5.2.

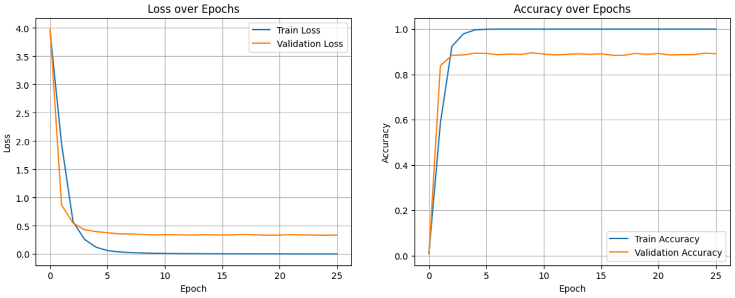

Figure 5: Training curves over 25 epochs with simultaneous fine-tuning for $l = 2$.

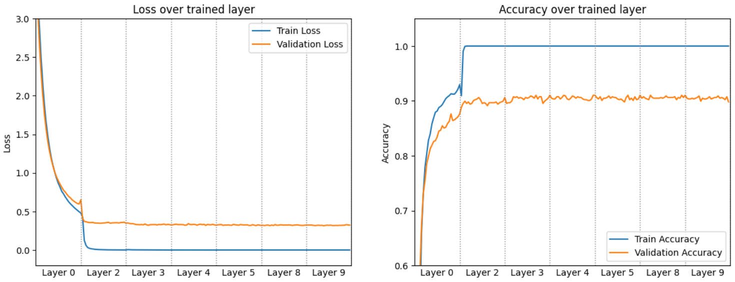

Figure 6: Training curves over all trained layers (each trained for 25 epochs) using gradual unfreezing.

The training times depended on the chosen value for l and wether all layers were considered for gradual un-freezing or not. In general they were similar. The results from the experiments indicate that only the last layers need to be trained since there is little to no effect in training the earlier layers.

This result is not a surprise since we are working with a pre-trained network and earlier layers express more general features.

To cope with the overfitting behavior of the network, data augmentation and L2 regularization were considered. The applied augmentation included random resized cropping with a scale between 80% and 100% of the original size, random horizontal flipping, and a random rotation of up to ±15 degrees with a probability of 50%. Tests for several values for λ have been carried out. Figure 7 shows the effect of data augmentation and L2 regularization for the gradual un-freezing strategy.

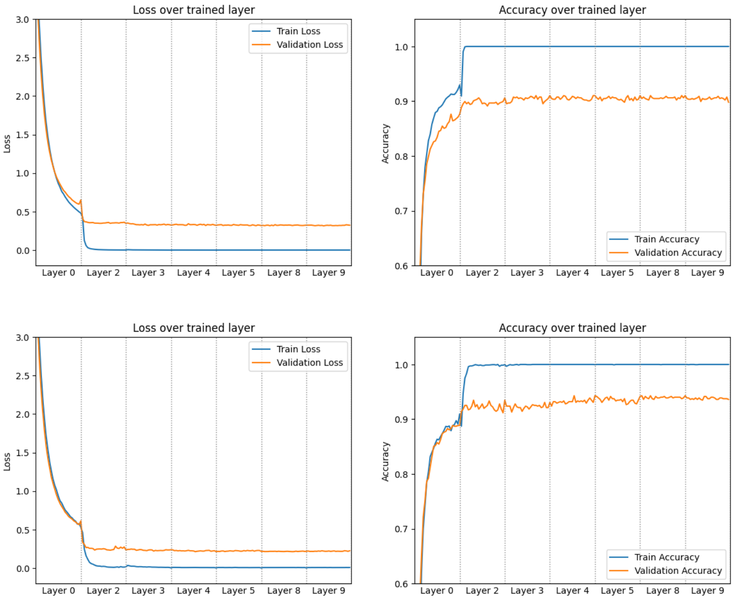

Figure 7: Comparison of training curves for gradual un-freezing without (top) and with (bottom) data augmentation and L2 regularization with $\lambda = 1e - 5$.

A decrease in overfitting can be observed as indicated by the lower validation loss. Compared to the improvements in performance by using data augmentation in the assignments the improvements observed for this network are rather small.
One possible explanation is the size of the training and validation datasets (Oxford-IIIT: 7,349 vs CIFAR-10: 60,000) and the fact that each image is only used once per epoch. Nevertheless data augmentation and L2 regularization were mostly retained throughout the project due to its positive effect on reducing overfitting.
Another reason for the overfitting behavior was based on the high number of epochs (`num_epochs = 25`) each layer was trained for. By considering fewer epochs and using ResNet50 instead of ResNet18 in order to maintain model complexity a reduction of overfitting as seen in Figure 8 and an improvement in test accuracy to **92.01%** was achieved. Since no significantly improvements could be observed for training earlier layers, only the last 5 layers were considered.

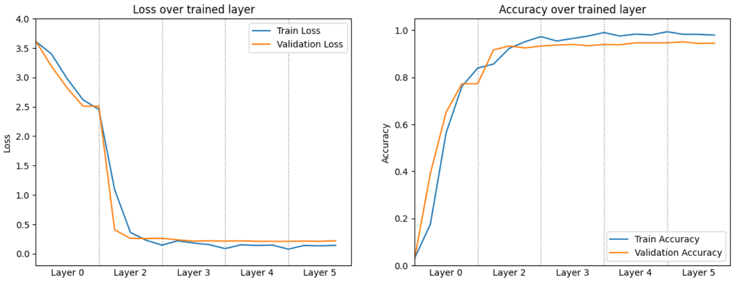

Figure 8: Training curves over all trained layers (each trained for 3 epochs) using gradual un-freezing on a pre-trained ResNet50.

A uniform grid search was done to investigate the affect of l2 regularization on network performance. The results are presented in 7. L2 regularization also helps reduce overfitting and produce better validation results.

λ	**Train. Acc.**	**Val. Acc.**
0.1	92.73%	90.08%
0.01	97.55%	94.84%
0.001	98.17%	94.57%
0.0001	98.23%	94.02%

Table 7: Effect of l2 regularization on network performance.

In some cases it might be advantageous to not fine-tune the batch norm layers of the network, especially when working on a small dataset. To evaluate this hypothesis a training run using the gradual un-freezing with data augmentation was carried out, where all batch norm layers have been skipped. In our experiments we did not find any benefit in skipping the batch norm layers.

At last a learning rate scheduling algorithm with an exponentially decreasing learning rate was implemented. The learning rate is determined by the initially defined learning rate η_0 and the current epoch and layer (starting from the last) via the equation:

$$\eta_{curr} = \eta_0 * \alpha^{layer} * \beta^{epoch} \tag{4}$$

Where $1 - \alpha$ defines the decrease of the learning rate per layer ($\alpha = 0.9 \rightarrow 10\%$ decrease per layer) and $1 - \beta$ the decrease per epoch. With this definition we derived three basic settings for the learning rate schedule: a constant learning rate (`const`) for $\alpha = \beta = 1$, a decreasing learning rate per layer (`layer_decr`) for $\alpha < 1$ and a decreasing learning rate per epoch (and layer) (`layer_epoch_decr`) for $\beta < 1$. Using a decreasing learning rate per layer ($\alpha = 0.8$) we were able to achieve the best final test accuracy of **92.37%**.

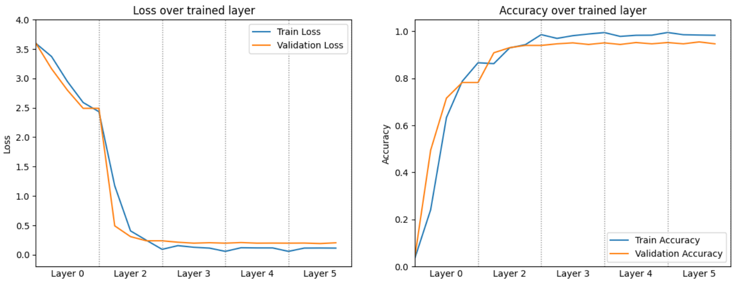

Figure 9: Training curves for the best trained network.

A similar approach was chosen to improve the performance when fine-tuning L layers simultaneously. The best results were achieved with the following set of hyperparameters: $\lambda = 0.001$, $\eta = 10^{-4}$, and data augmentation. In addition, a learning rate decay of 0.65 was applied after each epoch. The test accuracy achieved when training for 5 epochs with fine-tuning up to layer 2 was 92.64 %. The corresponding training plots are shown in 10.

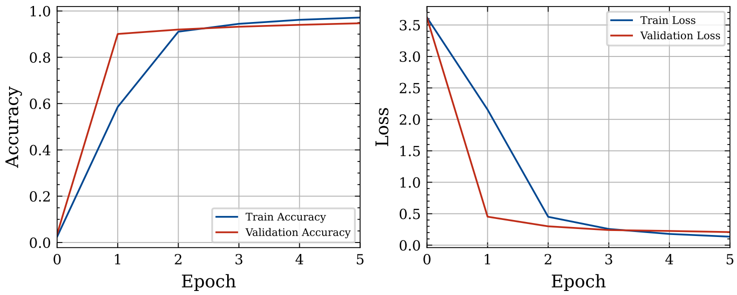

Figure 10: Training curves with simultaneous fine-tuning and optimized parameters

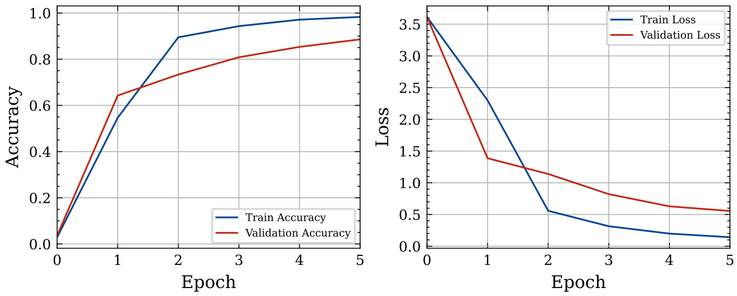

Figure 11: Training curves with imbalanced classes and without any compensation

This run also serves as a benchmark to test the robustness of the network with imbalanced classes. There, the number of images per cat breed was reduced to 20 % in the training data set, the resulting training curve is shown in 11. For compensation, in the first approach, a weighted cross-entropy loss is applied.

$$\mathcal{L}_{\text{WCE}} = -\sum_{i=1}^{37} w_i \cdot y_i \cdot \log(p_i) \tag{5}$$

This gives a test accuracy of 89.64 %, the resulting plots are shown in 12. For further evaluation, compensation is made by oversampling the cat classes. Therefore, a weighted sampler is used, which allows samples of classes with fewer elements to be drawn with a higher probability. It is also possible for the same samples to be drawn several times. With that approach, a test accuracy of 89.70 % can be achieved, the corresponding plots are shown in 13.

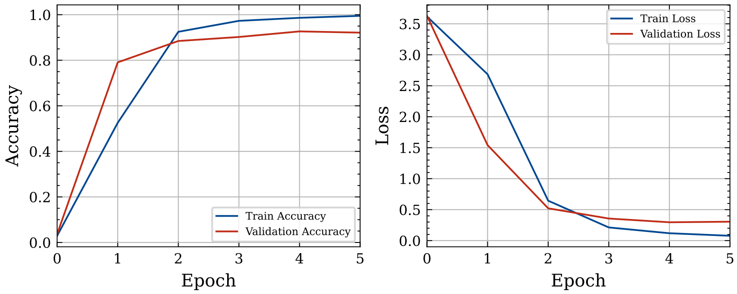

Figure 12: Training curves with imbalanced classes and weighted cross-entropy loss

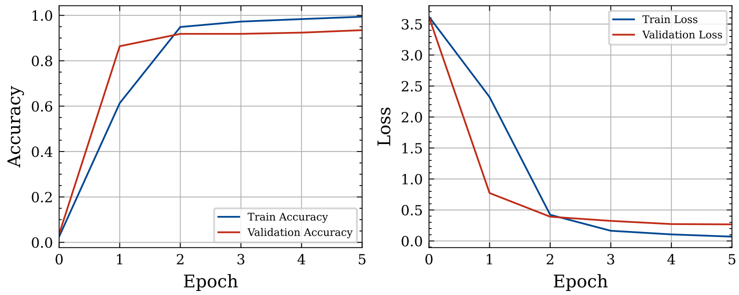

Figure 13: Training curves with imbalanced classes and over-sampling of the minority classes

A.3 FixMatch

To get a first impression of the performance using FixMatch we trained the entire ResNet50 network for different amounts of labeled images per class and wanted to use this result as a benchmark for later tests using gradual un-freezing and simultaneous training of the last layers. However the results in Table 8 present a bad performance, especially for larger numbers of images per class.

Images per Class	Labeled Data (%)	FixMatch (30 Epochs)
40	50%	72.97%
16	20%	60.17%
4	5%	41.38%

Table 8: Test accuracy with limited labeled data when training the whole network.

To investigate the reason for the bad performance we examined the Loss and Accuracy plots shown in Figure 14.

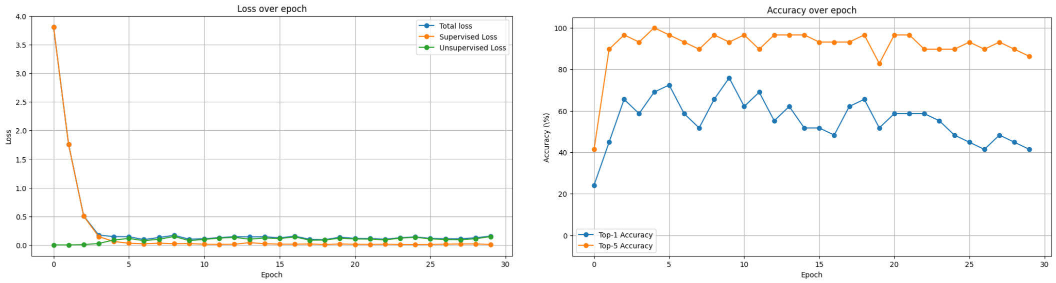

Figure 14: Loss and Accuracy over epoch using FixMatch when training the whole network.

The decreasing Top-1 Accuracy is particularly noteworthy. With the help of the plots we came to the conclusion that training of the whole network is not purposeful for the FixMatch Method. Therefore for the experiments we focused on simultaneous training of the last layers. The results we present in section 5.3 show promising results.

A.3.1 FixMatch Experiments with Fine Tunning

Below we present training curves of the experiments presented in 5.3 in Table 6. The left plot presents top-1 and top-5 validation accuracies over epochs. Note that top-5 accuracy is less meaningful here due to the limited number of classes. The right plot shows the training losses, including total loss and supervised and unsupervised components.

Due to limited computational resources, training with 20 and 50 labeled images per class was performed for only 5 epochs.

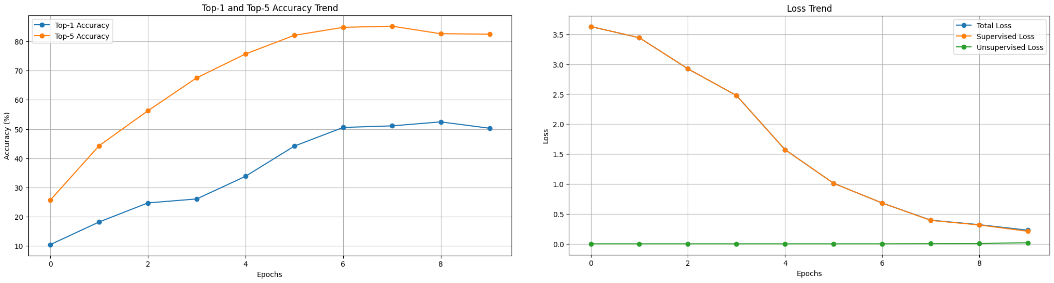

Figure 15: Accuracy and Loss Training curves for FixMatch 1 labeled images per class.

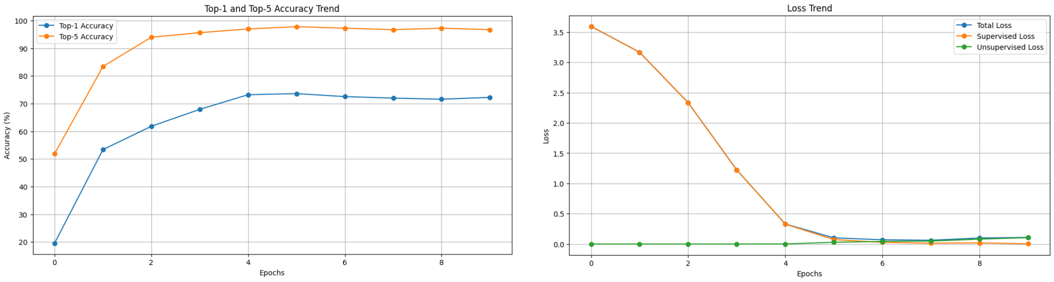

Figure 16: Accuracy and Loss Training curves for FixMatch 2 labeled images per class.

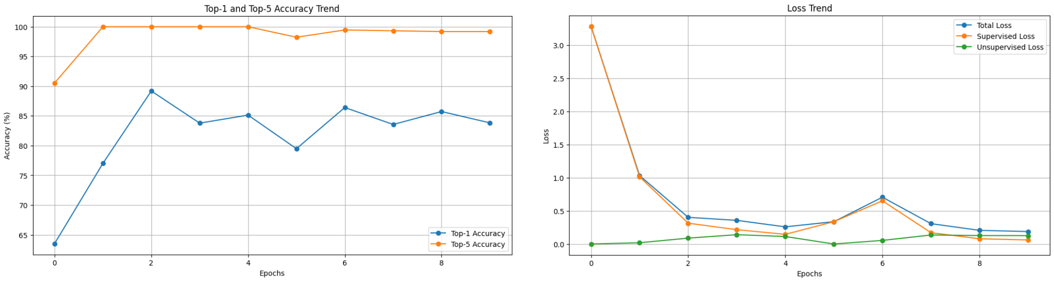

Figure 17: Accuracy and Loss Training curves for FixMatch 10 labeled images per class.

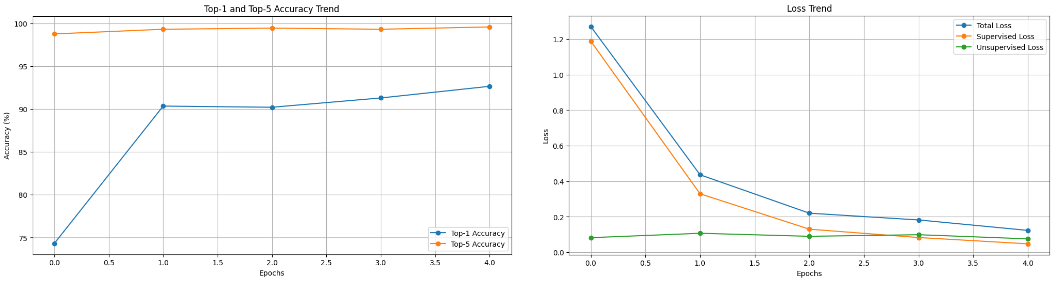

Figure 18: Accuracy and Loss Training curves for FixMatch 20 labeled images per class.

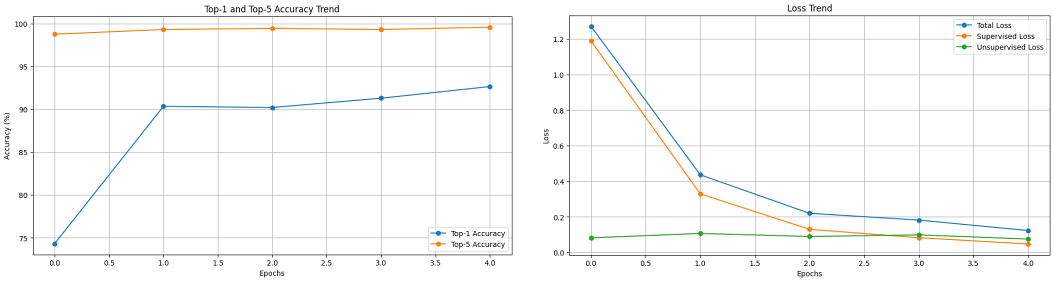

Figure 19: Accuracy and Loss Training curves for FixMatch 50 labeled images per class.