NATIONAL CHIAO TUNG UNIVERSITY

Institute of Computer Science and Engineering

Selected Topics in Visual Recognition using Deep Learning

Final Project

Authors:

0780828 - Alisher Mukashev

0845086 - Alessandro Saviolo

0756098 - Sophia Lien

25th December 2019

Summary

1. Introduction

1.1 Problem Statement

The iNaturalist2019 Challenge [20] is a large scale species classification competition sponsored by Microsoft [21]. This competition is part of the Fine-Grained Visual Categorization FGVC6 [18] workshop at the Computer Vision and Pattern Recognition Conference CVPR 2019 [19].

It is estimated that the natural world contains several million species of plants and animals. Without expert knowledge, many of these species are extremely difficult to accurately classify due to their visual similarity. The goal of this competition is to push the state-of-the-art in automatic image classification for real world data that features a large number of fine-grained categories.

1.2 The Importance of this Problem

With the advent of deep Convolutional Neural Networks the performances of image classification has witnessed incredible progress. The success is mainly addressable to available and high-quality large-scale datasets, such as ImageNet and MS COCO. However, in contrast with these datasets exhibiting roughly uniform distributions of class labels, real-world datasets always have skewed distributions with a long tail (i.e., few classes occupy most of the data, while most of the classes have rarely few samples - see Figure 1.1). Moreover, more and more long-tailed datasets reflecting the realistic challenges are constructed, such as iNaturalist.

Therefore, by solving this problem, we are able to efficiently solve image classifications tasks based on real datasets.

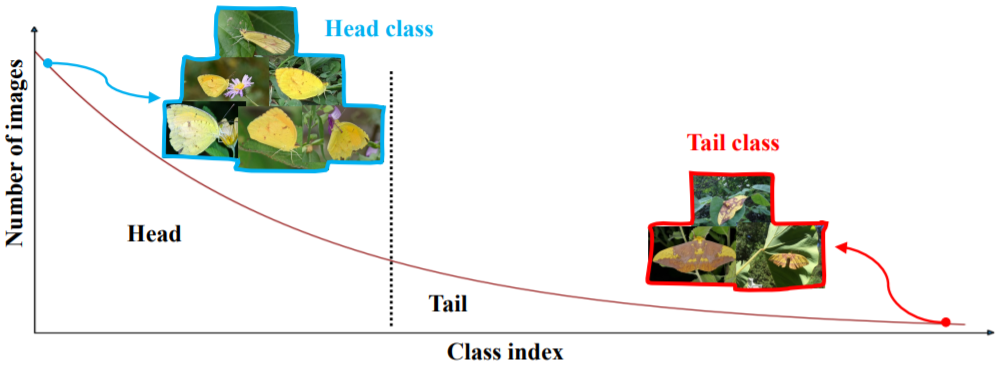

Figure 1.1: Distribution of a long-tailed dataset.

1.3 The Difficulty of this Problem

Real-world large-scale datasets often display the phenomenon of long-tailed distributions. The extreme imbalance causes tremendous challenges on the classification accuracy, especially for the tail classes. Moreover, iNaturalist dataset is not only imbalanced but also fine-grained. This means that the classifier must be accurate enough to classify different classes on tiny details.

Class re-balancing strategies can yield better classification performance for long-tailed problems. In this work, we reveal that the mechanism of these strategies is to significantly promote classifier learning but will damage the representative ability of the learned deep features.

To sum up, real-world datasets impose many challenges for classification tasks, and our contribution consists on combining different techniques to overcome such challenges. In Chapter 2, these techniques are introduced and discussed. In Chapter 3, the proposed approach is presented in details. Finally, in Chapter 4, the experimental results proving the effectiveness of our approach are illustrated and described.

2. Related Work

The proposed model is inspired by the iNaturalist2019 Challenge winning approach [1, 22]. In this chapter, we analyze and illustrate the winning model. Then, in Chapter 3 we present and discuss our approach to reproduce the model and our contribution to improve the model performances.

2.1 Transfer Learning

From a deep learning perspective, the image classification problem can be solved through *transfer learning*. Actually, several state-of-the-art results in image classification are based on transfer learning solutions [2–4]. With transfer learning, instead of starting the learning process from scratch, we start from patterns that have been learned when solving a different problem. This way we leverage previous learnings and avoid starting from scratch.

In Computer Vision, transfer learning is usually expressed through the use of pre-trained models. Accordingly, due to the computational cost of training such models, it is common practice to import and use models from published literature (e.g., VGG, Xception, ResNet). A comprehensive review of pre-trained models' performance on computer vision problems using data from the ImageNet [5] challenge is presented by [6].

2.2 Shift-Invariant Convolutional Neural Network

An important property of any model whose goal is to perform localization or segmentation is that the output of the model should be shift-invariant with regard to its input. However, as [15] has noted, standard CNNs use downsampling layers while ignoring sampling theorem, and are therefore not shift-invariant. To circumvent this problem, an anti-aliasing filter is required prior to every downsampling part of the network. In particular, [15] proposed the insertion of a blur kernel as a low-pass filter prior to each downsampling step in the network; and thereby demonstrated an increased accuracy across several commonly used architectures and tasks. Following [15], we thus modify the backbone of our base model and integrate such low-pass filters as part of our network.

2.3 Dealing with Long-Tailed Datasets

When dealing with long-tailed datasets, deep learning methods are not feasible to achieve outstanding recognition accuracy due to both the data-hungry limitation of deep models and also the extreme class imbalance trouble of long-tailed data distributions. In literature, the state-of-the-art methods for handling long-tailed problems are called *re-balancing strategies*. Such strategies can be divided in two groups: *re-sampling* [7–10] and *cost-sensitive re-weighting methods* [11–13, 15]. Re-sampling consists in over-sampling data for minority classes or under-sampling data for dominant classes. Re-weighting methods allocate large weights for training samples of tail classes in loss functions.

If we consider the learning process of a deep classification network and separate it into representation learning (done by the backbone network) and classifier learning (done by the last fully-connected layers), we can observe that these re-balancing strategies are mostly effective to promote the classifier learning, by directly influencing the classifier weights. But, these methods may also have some adverse effects by damaging the representative ability of the learned deep features (i.e., representation learning). In particular, when data imbalance is severe, re-sampling has the risk of over-fitting the tail data when over-sampling, or under-fitting the whole data distribution when under-sampling. For re-weighting, it will distort the original distributions by directly changing or even inverting the data presenting frequency.

In the paper [1] the authors proposed a unified Bilateral-Branch Network (BBN) model to take care of both representation learning and classifier learning. Also, in this paper a novel cumulative learning strategy is developed for adjusting the bilateral learnings and coupled with the BBN model's training.

3. Proposed Approach

3.1 Approach Overview

In our work we propose a unified Bilateral-Branch Network (BBN) model with anti-aliasing down-sample layer to take care of both representation learning and classifier learning. The model architecture is illustrated in Figure 3.1.

The proposed model consists of four main components. For the first part, we design two branches equipped with uniform sampling and reverse sampling, respectively. One is the conventional learning branch and the other one is the re-balancing brach. Both branches use the same residual network structure and share all the weights (except for the last residual block) to learn the data distribution representation. Then, two samples are fed into their own corresponding branch to acquire the feature vectors $\mathbf{f}_c$ and $\mathbf{f}_r$. We also use a specific cumulative learning strategy for shifting the learning attention between the two branches during the training phase. In other words, by controlling the weights for $\mathbf{f}_c$ and $\mathbf{f}_r$ with an adaptive trade-off parameter α, the outputs will be integrated together according to α by element-wise addition and the weighted feature vectors will be sent into the classifiers. For each class $i \in 1, 2, ..., C$, the softmax function calculates the probability of the class. Then, we use the cross-entropy loss function and the output probability distribution.

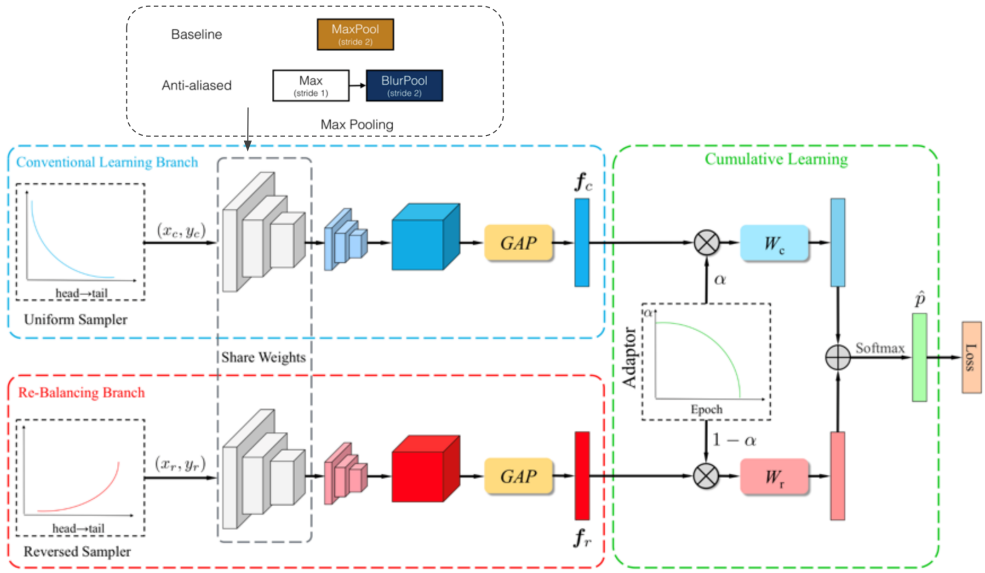

Figure 3.1: The proposed model. Note that GAP is sort for Global Average Pooling.

3.2 Details of Approach

3.2.1 Two sampling branches

The input data for the conventional learning branch comes from a uniform sampler, where each sample in the training dataset is sampled only once with equal probability in a training epoch. The uniform sampler retains the characteristics of the original distributions, and therefore benefits the representation learning. On the other hand, the re-balancing branch aims to alleviate the extreme imbalance and further improve the classification performance on tail classes, whose input data comes from a reversed sampler. For the reversed sampler, the sampling possibility of each class is proportional to the reciprocal of its sample size, (i.e., the more samples in a class, the smaller sampling possibility the class has).

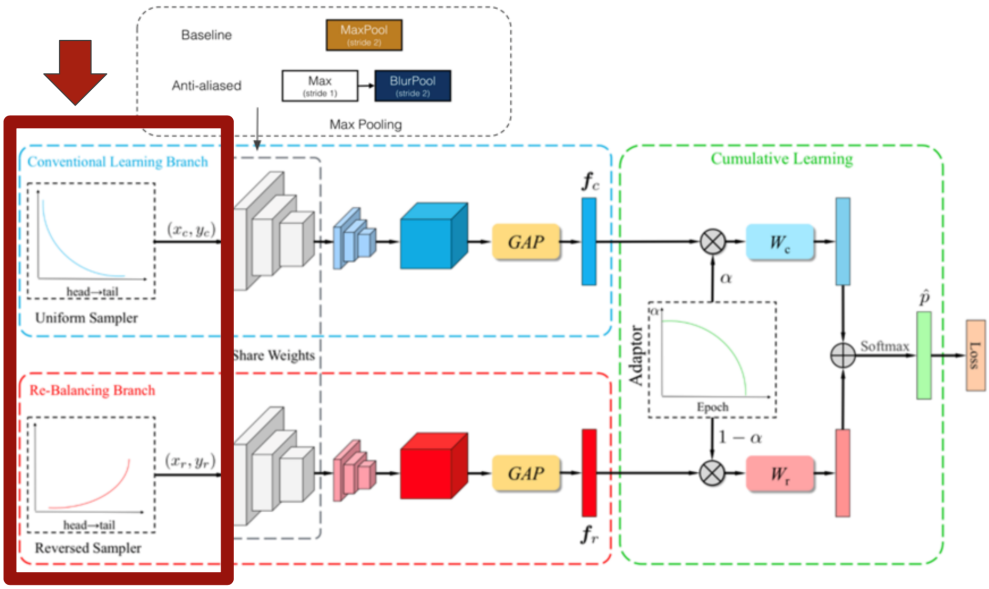

Figure 3.2: The two sampling branches implemented in the proposed model.

3.2.2 Representation Learning

In our model both branches economically share the same residual network structure as illustrated in Figure 3.3. We use ResNets-50 as our backbone network. Since [15] has noted that conventional methods for reducing spatial resolution (i.e., max-pooling, average pooling, and strided convolution) all break shift-equivariance, and in order to increase robustness and stability, we integrate anti-aliasing layer. For the Max-pooling, the Max operation preserves shift equivariance, as it is densely evaluated in a sliding window fashion, but subsequent subsampling does not. We simply add an anti-aliasing filter to combine blurring and subsampling (Figure 3.4).

Furthermore, two branch representation learning networks, except for the last residual block, share the same weights. There are two benefits for sharing weights: on one hand, the well-learned representation by the conventional learning branch can benefit the learning of the re-balancing branch; on the other hand, sharing weights will largely reduce computational complexity in the inference phase.

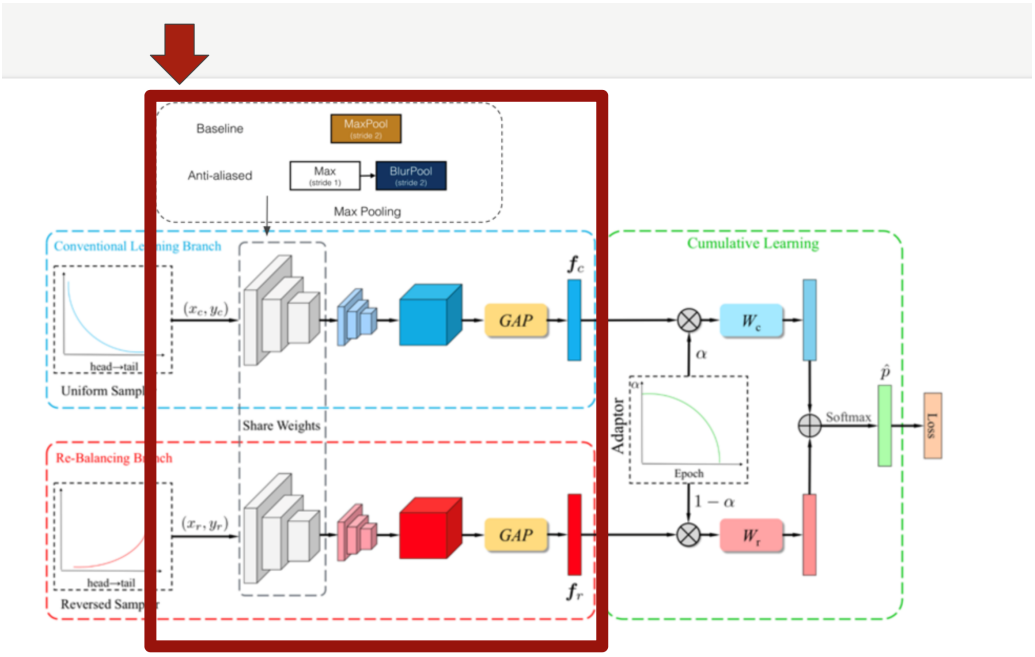

Figure 3.3: The Backbone architecture of the proposed model. This architecture is responsible for representation learning. Note the shared weights across the two branches.

3.2.3 Cumulative learning

Cumulative learning strategy is proposed to shift the learning focus between the bilateral branches by controlling the weights for features produced by two branches and the classification loss L. It is designed to first learn the universal patterns and then pay attention to the tail data gradually. In the training phase, the feature $\mathbf{f}_c$ of the conventional learning branch will be multiplied by α and the feature $\mathbf{f}_r$ of the re-balancing branch will be multiplied by $1-\alpha$, where α is automatically generated according to the training epoch.

$$\alpha = 1 - \left(\frac{T}{T_{max}}\right)^2,$$

In intuition, the adapting strategy for α is based on the motivation that the learning focus should gradually change from feature representations to classifiers, which can exhaustively improve long-tailed recognition accuracy.

$$\mathbf{z} = \alpha \boldsymbol{W}_c^\top \boldsymbol{f}_c + (1-\alpha)\boldsymbol{W}_r^\top \boldsymbol{f}_r,$$

With α decreasing, the main emphasis of model turns from the conventional learning branch to the re-balancing branch. Different from two-stage fine-tuning strategies, α ensures that both branches for different goals can be constantly updated in the whole training process, which could avoid the affects on one goal when it performs training for the other goal.

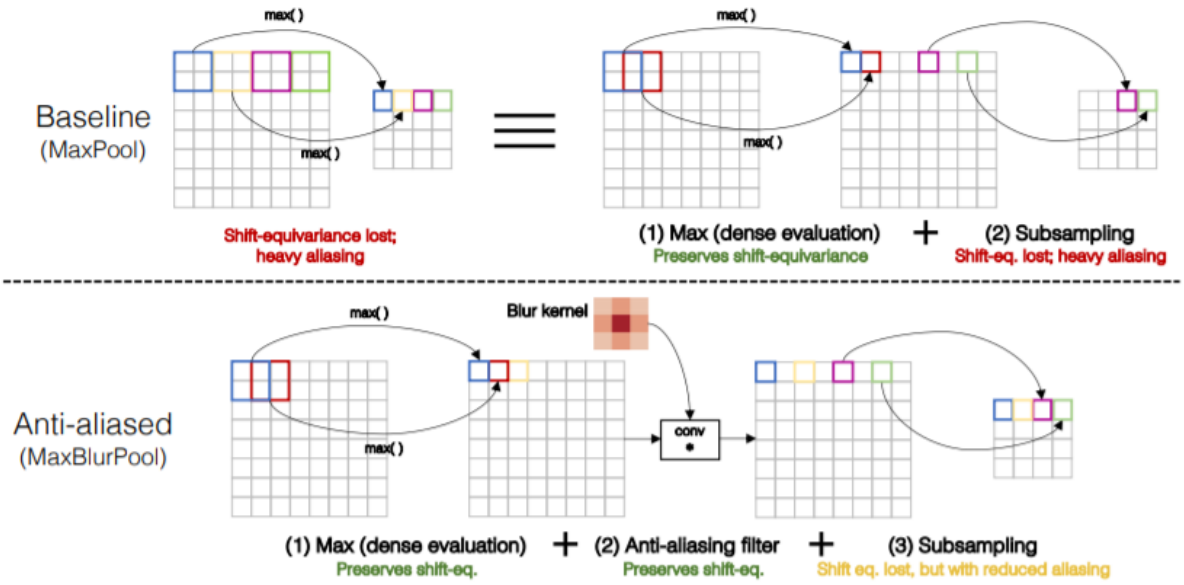

Figure 3.4: Anti-aliased max-pooling. In the top, Pooling does not preserve shift-equivariance. It is functionally equivalent to densely-evaluated pooling, followed by subsampling. The latter ignores the Nyquist sampling theorem and loses shift-equivariance. In the bottom, we low-pass filter between the operations. This keeps the first operation, while anti-aliasing the appropriate signal. Anti-aliasing and subsampling can be combined into one operation, which we refer to as BlurPool.

3.2.4 Classifier

The weighted feature vectors which integrated together according to α by element-wise addition will be sent into the classifiers. For each class $i \in 1, 2, ..., C$, the softmax function calculates the probability of the class. Then, we use the cross-entropy loss function and the output probability distribution.

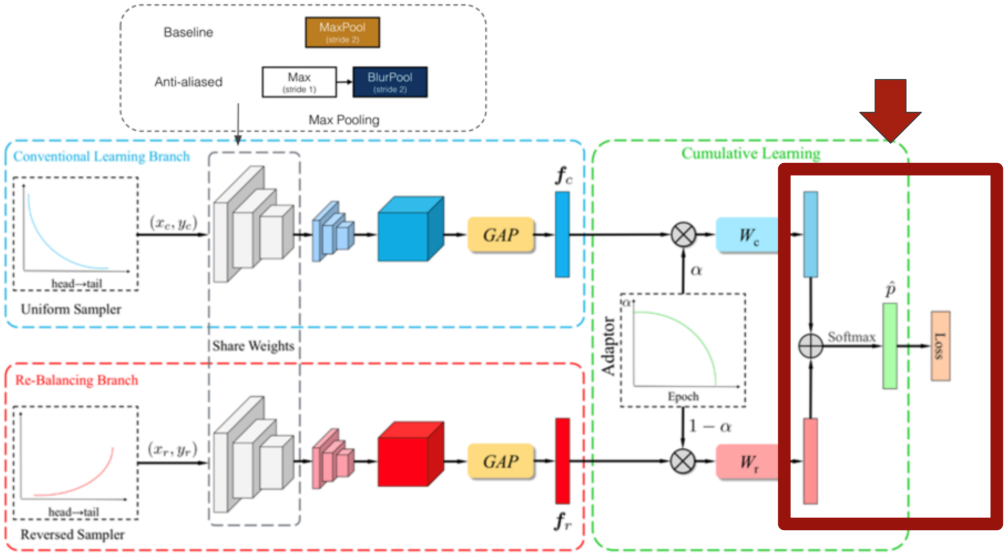

Figure 3.5: The classifier of the proposed model. This part of the model is responsible for classifier learning.

4. Experimental Results

4.1 Dataset and Metric for Evaluation

Previous versions of the challenge have focused on classifying large numbers of species. This year features a smaller number of highly similar categories captured in a wide variety of situations, from all over the world. In total, the iNaturalist2019 Challenge dataset contains 1010 species, with a combined training and validation set of 268243 images that have been collected and verified by multiple users from iNaturalist.

The 1010 classes have a long-tailed data distribution. This means that few classes occupy most of the data, while most classes have rarely few examples. In Figure 4.1 the classes distribution over the training set is illustrated.

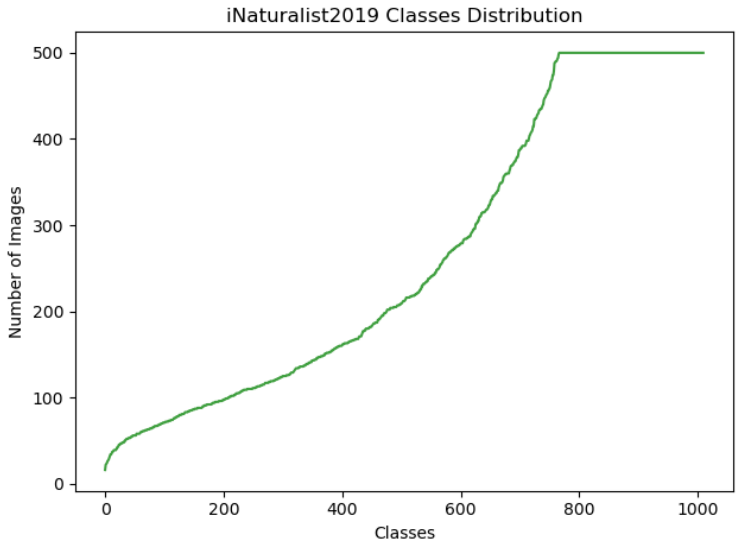

Figure 4.1: Classes distribution over the training set.

Top-1 error rate is used as evaluation metric. The model is considered to have classified a given image correctly if the target label is the model's top prediction. This is in contrast to the top-5 error rate where the model only needs to identify the correct label in the model's top 5 predictions.

4.2 Comparison with State-of-the-Arts

Our network is based on BBN, hence we provide comparison of BBN with the recently proposed LDAM [16] and CB-Focal [17] which achieve good classification accuracy on these long-tailed datasets.

Table 1. Top-1 error rates of ResNet-32 on long-tailed CIFAR-10 and CIFAR-100. (Best results are marked in bold)

Datasets	Long-tailed CIFAR-10			Long-tailed CIFAR-100		
Imbalance ratio	100	50	10	100	50	10
CE	29.64	25.19	13.61	61.68	56.15	44.29
Focal [17]	29.62	23.28	13.34	61.59	55.68	44.22
Mixup [32]	26.94	22.18	12.90	60.46	55.01	41.98
Manifold Mixup [28]	27.04	22.05	12.97	61.75	56.91	43.45
Manifold Mixup (two samplers)	26.90	20.79	13.17	63.19	57.95	43.54
CE-DRW [3]	23.66	20.03	12.44	58.49	54.71	41.88
CE-DRS [3]	24.39	20.19	12.62	58.39	54.52	41.89
CB-Focal [5]	25.43	20.73	12.90	60.40	54.83	42.01
LDAM-DRW [3]	22.97	18.97	11.84	57.96	53.38	41.29
Our BBN	**20.18**	**17.82**	**11.68**	**57.44**	**52.98**	**40.88**

Table 2. Top-1 error rates of ResNet-50 on large-scale long-tailed datasets iNaturalist 2018 and iNaturalist 2017. Our method outperforms the previous state-of-the-arts by a large margin, especially with 2× scheduler. "*" indicate original results in that paper.

Datasets	iNaturalist 2018	iNaturalist 2017
CE	42.84	45.38
CE-DRW [3]	36.27	40.48
CE-DRS [3]	36.44	40.12
CB-Focal [5]	38.88	41.92
LDAM-DRW* [3]	32.00	–
LDAM-DRW [3]	35.42	39.49
LDAM-DRW [3] (2×)	33.88	38.19
Our BBN	33.71	36.61
Our BBN (2×)	**30.38**	**34.25**

4.3 Ablation Studies

In our first attempt to reproduce the results declared by the paper [1], we have tried to use the same parameters listed in Table 4.1, but fixing the image size to 100×100. This choice is due to the computational challenge that is imposed by training such complex model from scratch. As a matter of fact, using such parameters the computational time required to train the model for each epoch is about 29 minutes.

The choice of reducing the image size has proved to be erroneous. The model starts overfitting after about 40 epochs. In Figure 4.2 the learning curves are illustrated. The private score obtained by using this model is 0.61603 and the public score is 0.61661.

(Hyper) Parameter	Value
Backbone	ResNet-50
Loss	CrossEntropy
# Max Epochs	180
Input Size	(224, 224)
Batch Size	128
Learning Algorithm	SGD
Initial Learning Rate	0.05
Momentum	0.9
Weight Decay	0.0001
Learning Rate Step	[120, 160]
Learning Rate Factor	0.1

Table 4.1: Parameters of the BBN model.

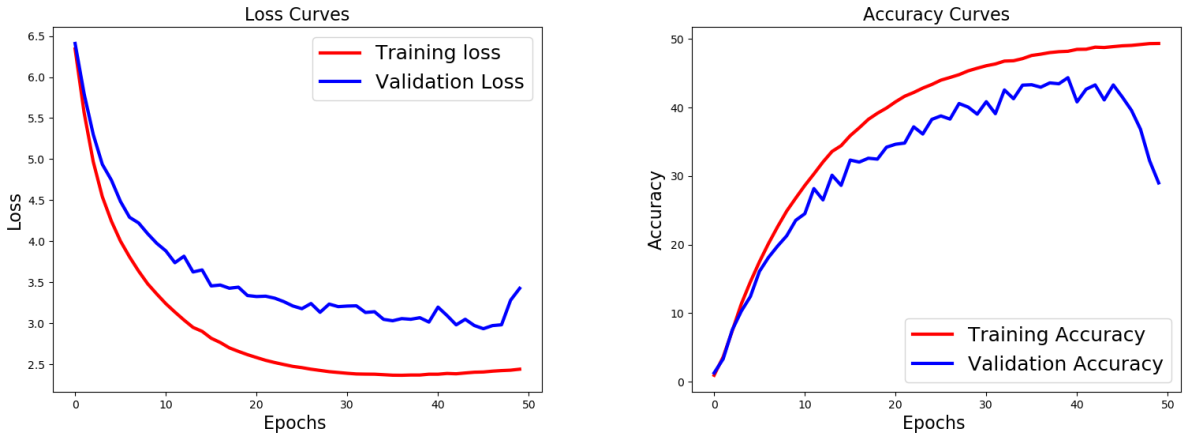

Figure 4.2: Learning curves produced by BBN model trained from scratch with the parameters specified in Table 4.1 and image size equal to 100×100.

Therefore, we have tried to double the image size, from 100×100 to 200×200, without changing any other paramater. Using this configuration, the computational time required to train the model for each epoch is about 45 minutes. Even if we doubled the size, the model is still overfitting after 50 epochs. In Figure 4.3 the learning curves are illustrated. The private score obtained by using this model is 0.48034 and the public score is 0.47838.

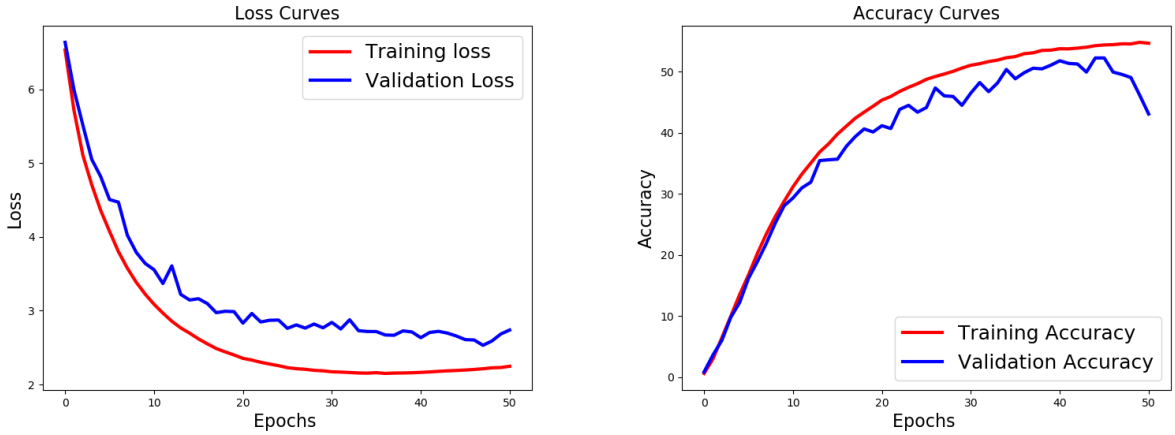

Figure 4.3: Learning curves produced by BBN model trained from scratch with the parameters specified in Table 4.1 and image size equal to 200×200.

We also have tried to play with the adaptive trade-off parameter α by changing the number of total training epochs. In particular, during training, we decay the learning rate at the 15th epoch by 0.1 (Figure 4.4).

We can see that by changing the adaptive trade-off parameter α showed much better results. Moreover, by decaying the learning rate we successfully reduced the overfitting effect.

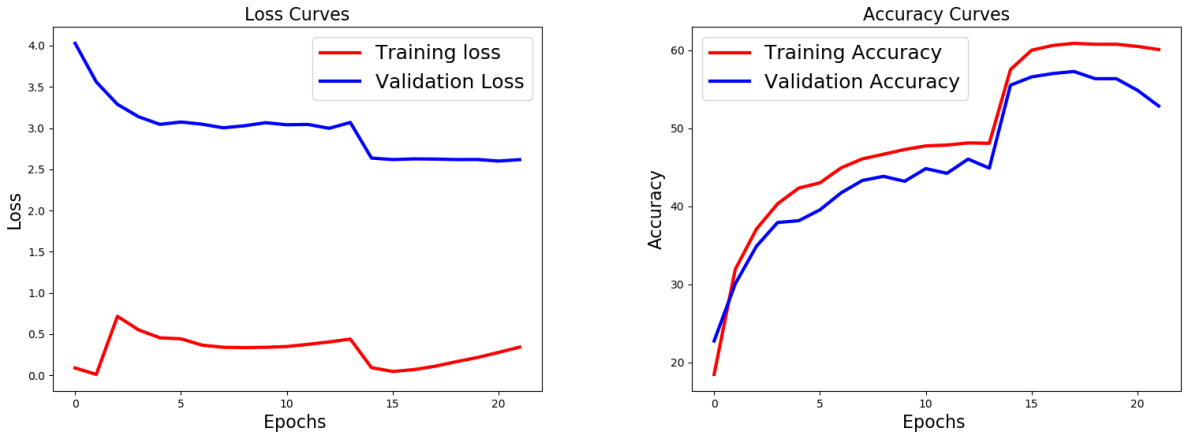

Figure 4.4: Learning curves produced by our model with the pretrained BBN weights, $T_{max} = 60$, with decaying learning rate.

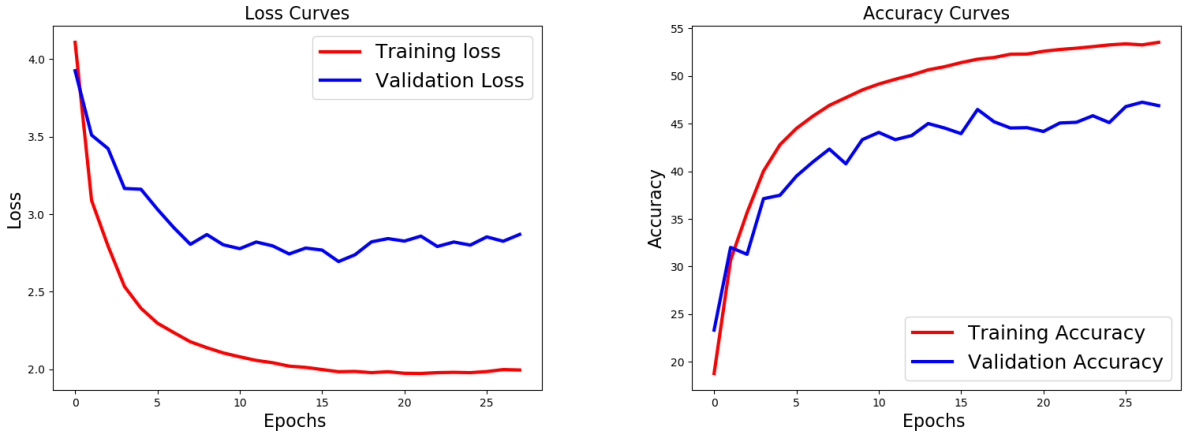

Figure 4.5: Learning curves produced by our model with the pretrained BBN weights, $T_{max} = 180$, with decaying learning rate.

To support the aforementioned studies, in Figure 4.6 and Figure 4.8 the learning curves produced by fixing, resp., α equal to 0 and to 1 are illustrated. Moreover, in Figure 4.7 and Figure 4.9 the submission scores obtained by the corresponding models are shown.

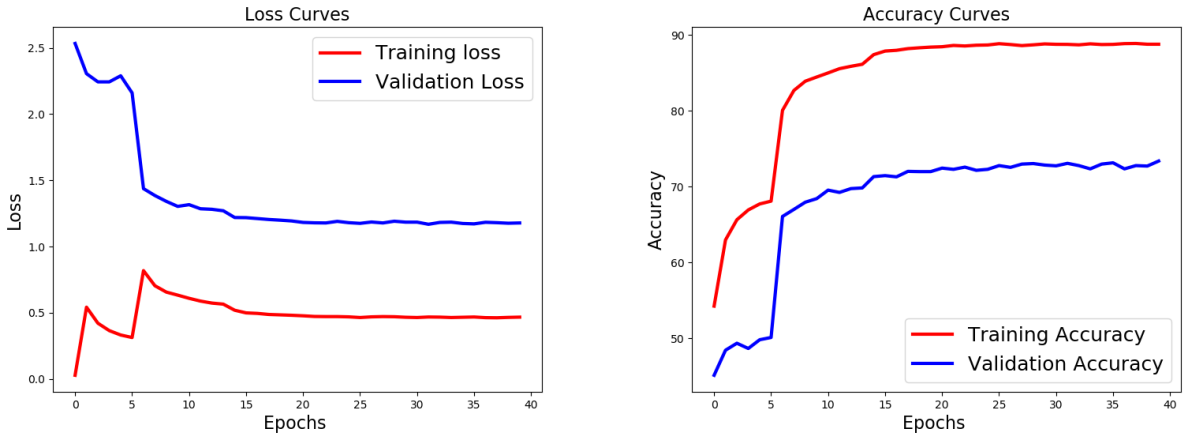

Figure 4.6: Learning curves produced by the model fixing α equal to 0.

Figure 4.7: Submission score obtained by training the model with α equal to 0.

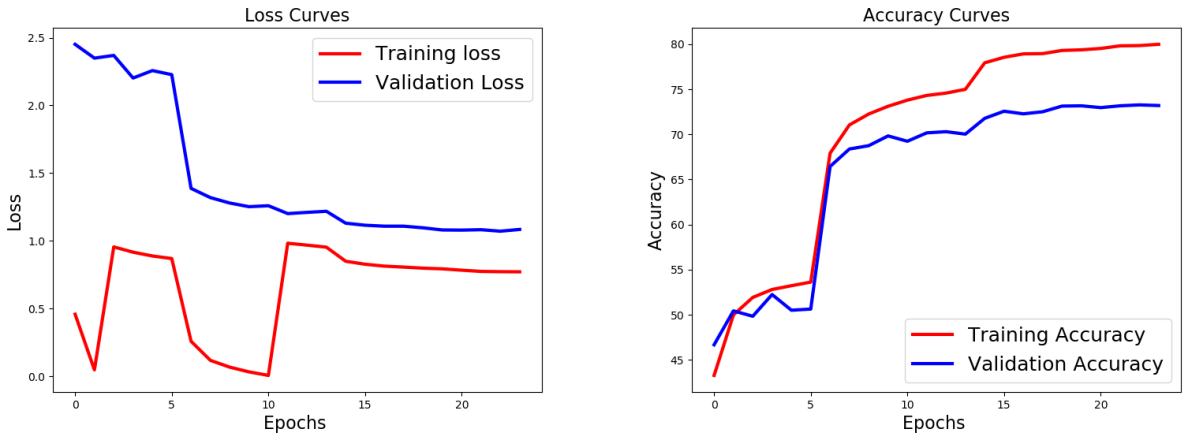

Figure 4.8: Learning curves produced by the model fixing α equal to 1.

Figure 4.9: Submission score obtained by training the model with α equal to 1.

The score obtained by the final model is shown in Figure 1. By comparing this score with the scores obtained by fixing α (Figure 4.7 and Figure 4.9), we can see that the model performs better with decreasing α.

5. Conclusions

We use bilateral branches for learning both representation and classification by using adaptive parameter α. With α decreasing, the main emphasis of model turns from the conventional learning branch to the re-balancing branch. Different from two-stage fine-tuning strategies, our model ensures that both branches for different goals can be constantly updated in the whole training process, which could avoid the affects on one goal when it performs training for the other goal. In order to increase robustness and stability, we integrate anti-aliasing layer with our backbone model.

This final project gave us opportunity to deal with big amount of data and imbalanced dataset. We learned how to train long-tailed dataset and use parameters which allows us to increase accuracy of training model. It was really new for us.

APPENDIX

A.1 Model Performance

Figure 1: Submission obtained by using the proposed model.

A.2 GitHub Link of the Code

The code of the project is available on GitHub at the repository [23].

A.3 Team Member Contribution

Literature Survey	0780828 (33%), 0845086 (33%), 0756098 (33%)
Approach Design	0780828 (30%), 0845086 (30%), 0756098 (40%)
Approach Implementation	0780828 (40%), 0845086 (30%), 0756098 (30%)
Report Writing	0780828 (30%), 0845086 (40%), 0756098 (30%)
Slide Making and Oral Presentation	0780828 (33%), 0845086 (33%), 0756098 (33%)

Table 1: Contribution of each team member to the development of this project. Note that the work burdain has been evenly distributed among the team members

Bibliography

[1] Zhou B., Cui Q., Wei X., and Chen Z. *BBN: Bilateral-Branch Network with Cumulative Learning for Long-Tailed Visual Recognition*. Megvii Research Nanjing, China.

[2] Krizhevsky A., Sutskever I., and Hinton G. *ImageNet Classification with Deep Convolutional Neural Networks*. University of Toronto, Canada.

[3] Simonyan K., and Zisserman A. *Very Deep Convolutional Networks for Large-Scale Image Recognition*. Computer Vision and Pattern Recognition, 2015.

[4] He K., Zhang X., Ren S., Sun J. *Deep Residual Learning for Image Recognition*. Computer Vision and Pattern Recognition, 2016.

[5] Deng J., Dong W., Socher R., Li L., Li K., and Fei-Fei L. *ImageNet: A Large-Scale Hierarchical Image Database*. Department of Computer Science, Princeton University, USA.

[6] Canziani A., Culurciello E., and Paszke A. *An analysis of deep neural network models for practical applications*. Purdue University, Indiana, USA, 2017.

[7] Chawla N., Hall L., Bowyer K., and Kegelmeyer P. *SMOTE: synthetic minority oversampling technique*. Journal of Artificial Intelligence Research, pp. 321–357, 2002.

[8] Byrd J., and Lipton Z. *What is the effect of importance weighting in deep learning?*. ICML, pp. 872–881, 2019

[9] Japkowicz N., and Stephen S. *The class imbalance problem: A systematic study*. Intelligent Data Analysis, pp. 429–449, 2002.

[10] Buda M., Maki A., and Mazurowski M. *Systematic study of the class imbalance problem in convolutional neural networks*. Neural Networks, pp. 249–259, 2018.

[11] Ren M., Zeng W., Yang B., and Urtasun R. *Learning to reweight examples for robust deep learning*. ICML, pp. 1–13, 2018.

[12] Cui Y., Jia M., Lin T., Song Y., and Belongie S. *Class-balanced loss based on effective number of samples*. CVPR, 9268–9277, 2019.

[13] Huang C., Li Y., Loy C., and Tang X. *Learning deep representation for imbalanced classification*. CVPR, pp. 5375–5384, 2016.

[14] Wang Y., Ramanan D., and Hebert M. *Systematic study of the class imbalance problem in convolutional neural networks*. NeurIPS, pp. 7029–7039, 2017.

[15] Zhang R. *Making convolutional networks shift-invariant again*. ICML, 2019.

[16] Cao K., Wei C., Gaidon A., Arechiga N., and Ma T. *Learning imbalanced datasets with labeldistribution-aware margin loss.* NeurIPS, pp. 1–18, 2019.

[17] Cui Y., Jia M., Lin T., Song Y., and Belongie S. *Class-balanced loss based on effective number of samples.* CVPR, pp. 9268–9277, 2019.

Sitography

[18] The Sixth Workshop on Fine-Grained Visual Categorization (FGVC6, 2019), last access 23 Dec. 2019, [Online].
https://sites.google.com/view/fgvc6/home

[19] Conference on Computer Vision and Pattern Recognition (CVPR, 2019), last access 23 Dec. 2019, [Online].
http://cvpr2019.thecvf.com/

[20] iNaturalist2019 Challenge (2019), last access 24 Dec. 2019, [Online].
https://www.kaggle.com/c/inaturalist-2019-fgvc6/overview/cvpr-2019

[21] Microsoft, last access 19 Dec. 2019, [Online].
https://www.microsoft.com/zh-tw/

[22] BBN Github Repository, last access 24 Dec. 2019, [Online].
https://github.com/Megvii-Nanjing/BBN

[23] Project Github Repository (2019), last access 25 Dec. 2019, [Online].
https://github.com/alishsuper/Visual_Recognition_Deep_Learning_final_project