CV703 Assignment 1
Multiple Fine-Grained Classification

Aidana Nurakhmetova
MBZUAI
aidana.nurakhmetova@mbzuai.ac.ae

Roba Al Majzoub
MBZUAI
roba.majzoub@mbzuai.ac.ae

Sultan Mobeen Abu Ghazal
MBZUAI
sultan.abughazal@mbzuai.ac.ae

Abstract

Fine-grained classification is still one of the challenging computer vision tasks, as it requires identifying subtle features which can describe an object at the subordinate level. In this report, our target is to present a solution for the fine-grained categorization of images from three benchmark datasets, namely, CUB-200-2011, Standford Dogs and FoodX and yield a performance higher than 77% for the concatenated dataset (CUB + Standford Dogs). We start with ResNetv2(50x1 bitm) baseline and implment a new model with a different architecture, with the highest achieved results of 81.1% of accuracy score.

1. Introduction

Human vision is one of the most complex systems in the world. It allows observers to distinguish between objects, animals, humans and other categories, locate them and find very specific contexts on the global and local levels. This makes the perception owner aware of his / her surrounding for better judgement of situations and better adaptation in the world. Within the computer vision domain we aim to mimic that human capability through implementing different systems that try to simulate the learning procedure and outcomes of that complex system, including those that tackle the classification of objects. Researchers aim to go deeper into the classification problem to implement in the systems the ability to further distinguish between entities within the same meta class, an approach known as "Fine Grained Classification".
This report discusses implementations of a set of models that aim to achieve a high accuracy on this task, and some modifications we introduced into those models.

2. Datasets

- Caltech's CUB-200-2011
- Stanford Dogs Dataset
- FoodX-251

Each of these datasets contains 200 or more fine-grained classes for each Meta class, from birds to dogs and even food. Caltech's CUB dataset is one of the datasets that are widely used for fine-grain classification, with 11788 images of birds out of which 5994 are for training and 5794 for testing. The set has in addition to its labels annotations for 15 parts, a bounding box and 312 attributes per image [6].
Standford's Dogs dataset consists of images from 120 different classes of dog breeds worldwide. It was created using ImageNet dataset particularly for fine-grained object categorization. Total number of images is 20580 and annotations provide not only labels, but bounding boxes, head RoI and pixel-level trimap segmentations too [3]. As for the last dataset, we used FoodX which has 251 fine grained classes, with 158000 images split into 118000 for training and 40000 for validation and testing [2]. However only labels were used for all three datasets without any other annotations or bounding box ground truths. The datasets trained on were CUB, a concatenation of CUB and dogs and foodX. A number of experiments were performed to decide on the baseline network and to improve the fine-grained classification of our proposed model

3. Experiments

The first part of the experiments focused on deriving the best baseline model for which we tried ResNet34, ResNet50, ResNet101, Deit (Data-efficient transformer), ViT small, SEResNet50, NasNet and other baselines.

Training on the three variants of Resnet (34, 50 and 101) were performed with the following settings: Batch size = 64, Dropout = 0.5, Epoch = 30, with simple RandomHorizontalFlip(), RandomVerticalFlip(0.1), RandomAffine(45, (0.3, 0.3), (2, 2)) for data transforms with Adam optimizer and a learning rate of LR = 1e-4.
Out of all the models we trained the ResNetv2(50x1 bitm) model was the one that performed best on the validation sets. We also tested the model on the three datasets mentioned above in the following manner: CUB dataset, a concatenated dataset formed of birds and dogs data, and the FoodX dataset. Since the main goal of this report work is to solve the classification task on the concatenated dataset with a good accuracy, our aim was to acquire the highest score on it. According to the tables shown in Table 1 and Table 3, the highest performance was achieved with the **ResNetv2(50x1 bitm)** which is considered as our baseline. Another aspect of the experiments was to overcome a problem of overfitting for which we applied several data augmentation techniques and added a dropout to the final layer. Table 1 shows the effects of these techniques on different variants of ResNet network. For all the models, only the final layer was fine-tuned to extract features from the pre-trained models, thus the previous layers were all frozen to avoid updating the parameters during optimization.
Optimizers: Different optimizers were used in the different training experiments we conducted. SGD and Adam variants were used since they have proven to yield excellent results in fine-grained classification.
Learning Rate: Different learning rates between 1e-3 and 1e-4 were used for training. A learning weight decay was used to decrease the learning rate every number of epochs.
Batchsize: The higher batch size was set to 128 which made convergence faster. Different batch sizes were tested, higher was better as seen in Table 3.

Table 1: Results of Resnet34, 50, 101 variants on CUB-200-2011.

Model	Params	Train Acc	Val Acc
Resnet34	365,256	48.6%	33.5%
Resnet50	102,600	54.37%	33.6%
Resnet101	409,800	59.2%	40.6%

4. ResNetv2

Resnet is one of the top models used for image classification due to the fact that it has the ability to utilize deeper models without the problems of degradation or vanishing gradients, which was solved by skip connections and identity blocks [1]. Another variant of the architecture was introduced by the authors of the original ResNet as ResNetv2(Fig. 1), which had a change in the arrangement of the layers and changes in normalization schemes in the residual block to achieve better results than its predecessor[4]. We fine-tuned our base model by freezing all the layers except the last layer of the network, the "ClassifierHead". This head which is composed of an average pooling layer and a convolutional layer whose output are then flattened to obtain a one dimensional vector output with the number of classes as its dimension. The train and validation accuracy of the baseline are shown in the results section. We used a training batch size of 128, a learning rate of 0.1e-4, and AdamW was used for optimization.The network was fine-tuned for 50 epochs.

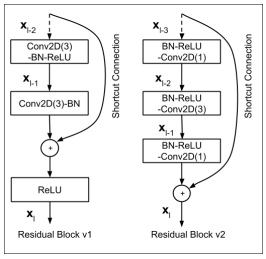


Figure 1: Different ResNet Bottleneck Architectures

4.1. Architecture Manipulations

For a good fine-grained classification, fine details are very important for the network. These details may help it learn more salient features for each class. We decided to implement downsampling skip connections from the output feature maps of the first 3 layers and concatenate them with the output of the final layer. (Note that one layer of this architecture is formed of multiple layers of convolutions, group normalizations and activation functions). Architecture is depicted in Fig.2.

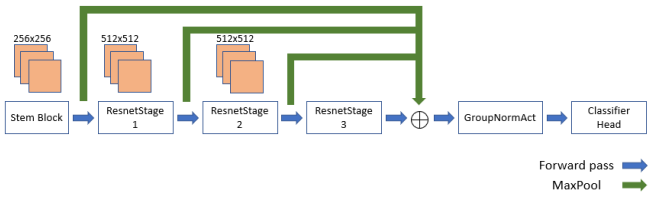


Figure 2: Proposed model

4.2. Experiment with Proposed model

Simple data augmentations were applied such as RandomHorizontalFlip(), RandomVerticalFlip(0.1) and RandomRotation(45). The data was resized to (224, 224) and normalized with the standard values from PyTorch. The model was trained for 50 epochs and learning rate and optimizer were same as those of baseline.

Table 2: Results with different models run over 50 epochs on CUB-200-2011 dataset.

Model	Batch	Augm	Optim	Val Acc
Resnet101	64	✓	SGD	55.5%
DeiT-16	64	✓	AdamW	71.2%
Resnetv2	32	✓	AdamW	71.7%
Resnetv2	64	✗	AdamW	75.3%
Resnetv2	128	✗	AdamW	75.6%
SEResnet50	128	✗	Adam	71.6%
NasNet	64	✗	Adam	54.6%

We also introduced a method of data augmentation by adding shadows to the input images where a set of translucent shadows were predefined and cast on the input images. The predefined shadows roughly resemble shadows cast by trees, leaves, branches, and poles which are some of the most common naturally occurring shadows. Along with the shadowing, a random horizontal flip is also applied. Shadowing augmentations were inspired from the paper [5].

Initially, the shadow augmentation was tested on a basic ResNet50 where it showed a small improvement in accuracy (<.5%).

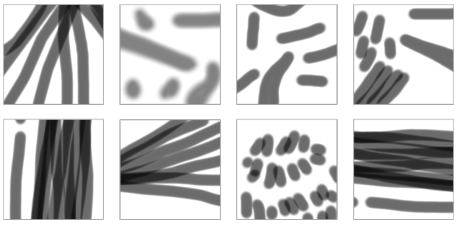
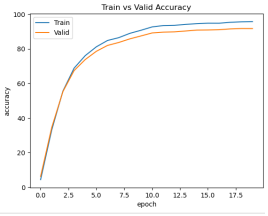


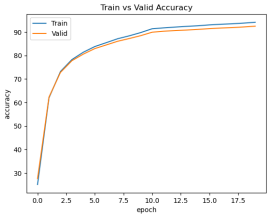
Figure 3: Introduced shadowing augmentation

4.3. Results and Discussion

The validation and test accuracy outputs can be seen in Table 4 for all datasets. The accuracy graphs for CUB-200-2011 are visualized in Figures 4a and 4b. The larger batch size helped baseline to improve the performance and augmentation helped reduce overfitting (Table 3). The accuracy



(a) Resnetv2 illustrated for 20 epochs on CUB-200-2011 dataset.



(b) Resnetv2 illustrated for 20 epochs on concatenated dataset.

Table 3: Comparison of the results obtained from the baseline and the proposed model on the concatenated dataset.

Model	Shadow Aug	Val Acc	Test Acc
Baseline	✗	78.3%	81.13%
Baseline	✓	65.9%	66.7%
NewModel	✗	82.4%	78.1%
NewModel	✓	59.0%	-

graphs for all dataset are illustrated in Figure 5. In all these graph, training models overfit, although the validation curve follows the same trend as the train accuracy curve, which is a sign of feature learning, and the test accuracy also shows promising results for further possible tuning of the network.

Proposed Model:

We attempted transfer learning through fine-tuning and feature extraction on the proposed architecture. With fine-tuning all layers from the pre-trained model were updated through back propagation, so the model had 26,786,152 parameters. For last one or two layers fine-tuning , the model had only 1,236,000 parameters which can help the network to train and converge faster. The datasets' sizes were not large enough (for both dog and cub) so getting good results for a whole-network fine tuning was not possible. For the shadow augmentation , it did not achieve such improvement on our model. This is due to the predefined shadows having low opacity which unintentionally occluded the object. We think this could be improved by creating shadows more carefully or by extracting shadows from samples from the same dataset. Using feature concatenation in the model showed an increase in the validation accuracy from that of the baseline, however the test accuracy decreased, which may show that the connections proposed are not helping the network to further generalize on the data. This can be due to the extreme maxpooling on feature maps.

Table 4: Baseline ResNetv2(50x1 bitm) accuracy results

Dataset	Valid Acc	Test Acc
CUB-200-2011	78.4%	78.3%
CUB + Dogs	78.3%	77.3%
FoodX Dataset	54.5 %	59.9%

4.4. Conclusion

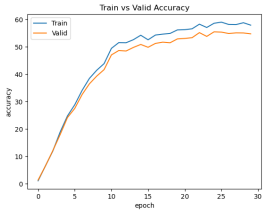
In this report, we worked on fine-grained classification of 3 different meta classes. ResNetv2 acted as the baseline model, and further experiments were conducted by implementing a new augmentation technique and effects of concatenating earlier features with later ones. This enabled extracting both global and local features, to try and focus on finer information by aggregating features from the preced-

ing layers at the final classification head.

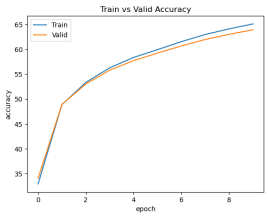
5. Appendix

Below accuracy graphs can be seen for Resnet34, 101 models and ResNetv2 on FoodX dataset.

(a) Resnet34



(b) ResNet101



(c) Resnetv2 illustrated for 10 epochs on FoodX dataset.

Figure 5: Train and validation accuracy graphs for ResNetv2

References

[1] Kaiming He, Xiangyu Zhang, Shaoqing Ren, and Jian Sun. Identity mappings in deep residual networks, 2016.

[2] Parneet Kaur, Karan Sikka, Weijun Wang, Serge Belongie, and Ajay Divakaran. Foodx-251: A dataset for fine-grained food classification, 2019.

[3] Aditya Khosla, Nityananda Jayadevaprakash, Bangpeng Yao, and Li Fei-Fei. Novel dataset for fine-grained image categorization. In *First Workshop on Fine-Grained Visual Categorization, IEEE Conference on Computer Vision and Pattern Recognition*, Colorado Springs, CO, June 2011.

[4] Ngo Le Huy Hien and Nguyen V.H. Recognition of plant species using deep convolutional feature extraction. 11:904–910, 06 2020.

[5] Osama Mazhar and Jens Kober. Random shadows and highlights: A new data augmentation method for extreme lighting conditions. *CoRR*, abs/2101.05361, 2021.

[6] C. Wah, S. Branson, P. Welinder, P. Perona, and S. Belongie. The Caltech-UCSD Birds-200-2011 Dataset. Technical report, 2011.