**Tugas Deep Learning – Computer Vision**

Nama: Anderies

NIM: 2240108910

Kasus: OXFORD-IIIT PET

Sumber Dataset: **DOI:**[10.1109/CVPR.2012.6248092](https://doi.org/10.1109/CVPR.2012.6248092)

Link Dataset: <https://www.kaggle.com/tanlikesmath/the-oxfordiiit-pet-dataset>

Model: ResNet50, EfficientNet-B7

**About Dataset:**

The Oxford-IIIT-Pet is covering 37 different breeds of cat and dog. There are 200 images for each breed class. With total of 7349 Images. The whole Oxford-IIIT-Pet classes and sub classes seen in Figure 1. This type of problem is fine grained object categorization [1].

People devote a lot of attention to their domestic animals, as suggested by the large number of social networks dedicated to the sharing of images of cat and dogs: Pet Finder, Catster, Dogster, My Cat Space, My Dog Space, The International Cat Association and several others. In fact, it is not unusual for owners to believe and post the incorrect breed for their pet, so having method of automated classification could provide a gentle way of alerting them to such errors. In real field making computers able to see the difference between pets is very advantageous for analysis across a variety of purposes in the industry as well as education and CCTV implementation for finding lost pets.

A collage of a cat and dog

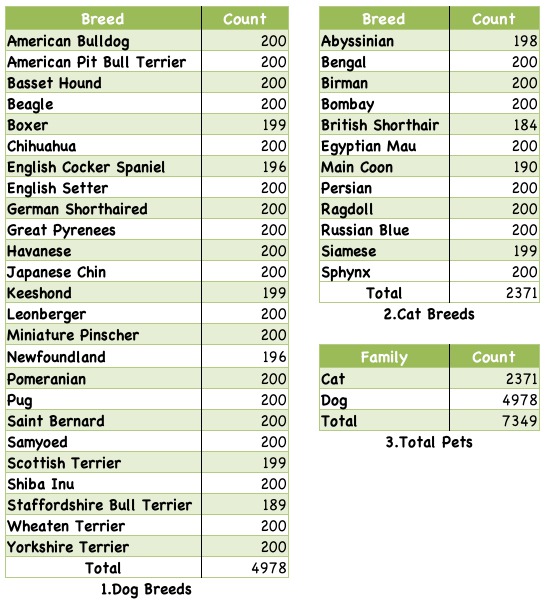
Description automatically generated with low confidence

Figure 1 Oxford-IIIT PET Class and Sub Classes

Dataset in Figure 1 then feed into two model ResNet [2] and EfficientNet-B7 [3] as comparison between state-of-the-art convolution model and deep networks convolution.

**Experiment:**

The implementation is carried out using the pytorch and python code libraries as programming languages.for using robust computer vision models you can visit this link <https://pytorch.org/vision/stable/models.html> to find available built-in model in pytorch, for the efficientNet-B7 model author using <https://github.com/lukemelas/EfficientNet-PyTorch>

Both model is using the same hyperparameter configuration and has been pretrained in ImageNet so the comparison can be apple to apple, configuration is not aiming for the best accuracy in solving dataset but aiming for checking performance of the models in a dataset. The configuration of model is as follows:

1. Optimizer SGD
2. Learning Rate 0.001
3. Momentum 0.9
4. Other parameters such as linear-decay is recommended default.

Table 1. ResNet-50 Pretrained with ImageNet

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| Epoch | Training | | Validation | |
| Loss | Accuracy | Loss | Accuracy |
| 1 | 1.8192 | 0.4843 | 0.5652 | 0.8378 |
| 5 | 1.0988 | 0.6814 | 0.6149 | 0.8505 |
| 10 | 0.5759 | 0.8295 | 0.2805 | 0.9252 |
| 15 | 0.5093 ( 0,066 ) | 0.8535 ( Differ ) | 0.2709 | **0.9422** |
| 19 | 0.4903 ( 0,019) | 0.8619 ( Differ ) | 0.2544 | 0.9365 |
| **20** | **0.4561** ( 0,034 ) | **0.8680** ( Differ ) | **0.2350** | 0.9379 |

Table 2. EfficientNet-B7 Pretrained with ImageNet

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| Epoch | Training | | Validation | |
| Loss | Accuracy | Loss | Accuracy |
| 1 | 2.2800 | 0.4522 | 1.2492 | 0.8590 |
| 5 | 0.7183 | 0.7989 | 0.3191 | 0.9168 |
| 10 | 0.5461 | 0.8424 | 0.2269 | 0.9351 |
| 15 | 0.5195 | 0.8523 | 0.2128 | 0.9337 |
| 19 | 0.5036 ( Differ ) | 0.8520 ( Differ ) | 0.2086 | 0.9337 |
| 20 | 0.5205 ( Differ ) | 0.8478 ( Differ ) | 0.2134 | 0.9323 |

In Table 1 and Table 2

Analysis of the jump loss and accuracy in ResNet-50 is better and more significant than that of EffecientNet-B7 in the range of epoch 10 to 20. EfficientNet-B7 tends to be more static with the difference in average accuracy and the loss on EfficientNet-B7 tends to be static in the range of 0.00 N and on ResNet 0.0N.

Final Result:

1. ResNet50

Training Complete in 49 Minutes 41seconds

Best validation accuracy was 94,22%

Testing Loss: 0.5606, Accuracy: 0.8480

2. EfficientNet-B7

Training Complete in 215minute 8s

Best Validation Accuracy 93,5%

Testing Loss: 0.4755 Acc: 0.8621

**Training Speed Result**

In Final Result above although ResNet50 is more fast in training speed than efficientNetB7, other trusted researchs showing contradiction of that [3]. And author conclude that google colab unstable computing power result can’t be adjustment of this researchs. In several previous studies such as VGG [4] vs ResNet training speed, it was written in a published journal that ResNet's speed was 8x faster, but in some experiments that occurred at Google Colab, the result was opposite and inconsistent, so training speed result in using google colab can’t be benchmark on how fast your models. Theoretically EfficientNet- B7is actually is more fastest than ResNet50.

**Accuracy Result**

In Final Result we can see in validation accuracy 94.22% by ResNet is outperform EfficientNet-B7 by 0,7% but the testing accuracy result shows that EfficientNet B-7 perform 86.21% and ResNet is 84.80%. both outperform resnet in loss and accuracy. And as for quite significant differences of training accuracy and validation accuracy causes by amount of training and validation number of images, in Training there are 80% of 7349 and In Validation there are 10% of 7349.

**Result Analysis**

The results of the research model pretrained efficientNet 86% testing accuracy and ResNet 84% testing accuracy. In Figure 2 and 3 seen on the elbow pattern, the result is not the best of global minima but rather good results on local minima of the pretrained model. And compared to validation accuracy the model is overfit and can be improved using dropout or regularization technique in upcoming implementation

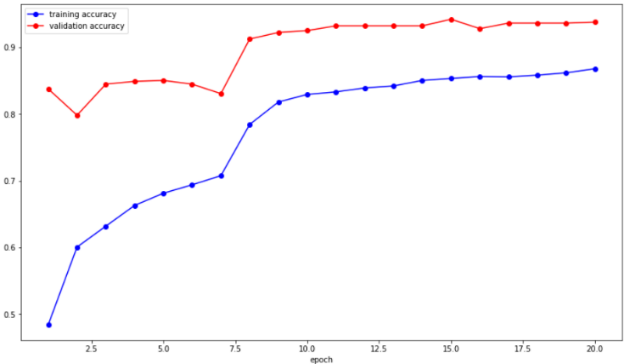
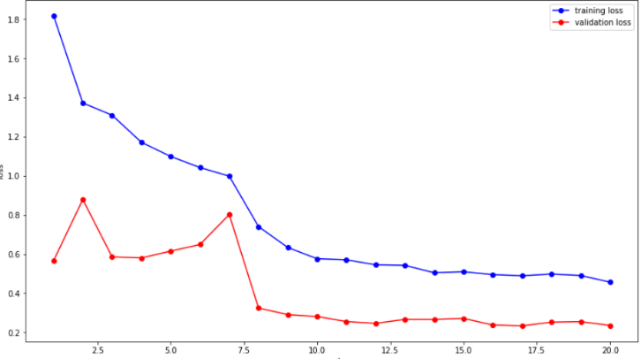


Figure 2. Resnet Model (Left) Accuracy Result, (Right) Loss Result

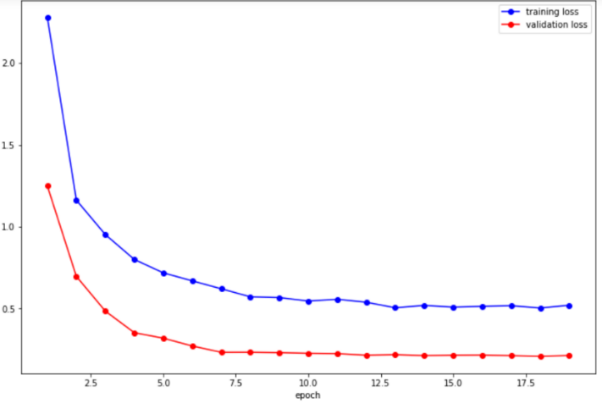
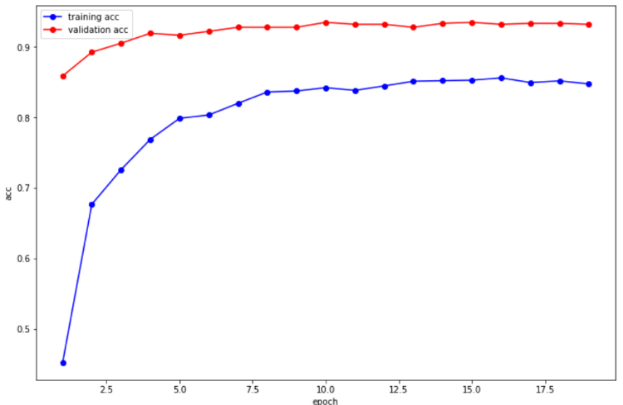
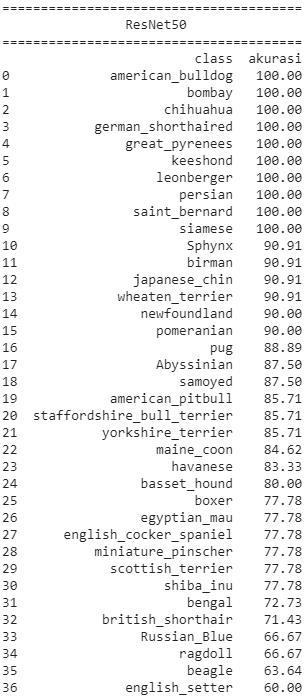
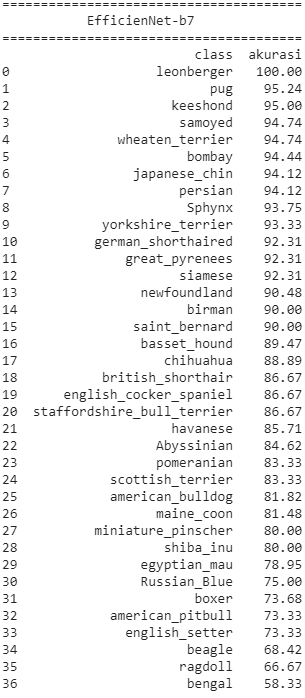


Figure 3. EfficientNet B-7 Model (Left) Accuracy Result, (Right) Loss Result

Figure 4 Each Class Accuracy of ResNet50 and EfficientNet-B7



Find Implementation of Author Code in github link below :

https://github.com/Anderies/tugas\_deep\_learning.git

**References**

[1] O. M. Parkhi, A. Vedaldi, A. Zisserman, and C. V Jawahar, “Cats and dogs,” in *2012 IEEE conference on computer vision and pattern recognition*, 2012, pp. 3498–3505.

[2] K. He, X. Zhang, S. Ren, and J. Sun, “Deep residual learning for image recognition,” in *Proceedings of the IEEE conference on computer vision and pattern recognition*, 2016, pp. 770–778.

[3] M. Tan and Q. V Le, “Efficientnet: Rethinking model scaling for convolutional neural networks,” *Int. Conf. Mach. Learn. (pp. 6105-6114)*, 2019.

[4] K. Simonyan and A. Zisserman, “Very Deep Convolutional Networks for Large-Scale Image Recognition,” *arXiv Prepr. arXiv1409.1556*, Sep. 2014, [Online]. Available: http://arxiv.org/abs/1409.1556.