Final project report

Selected Topics in Visual Recognition in Deep Learning

25.12.2019

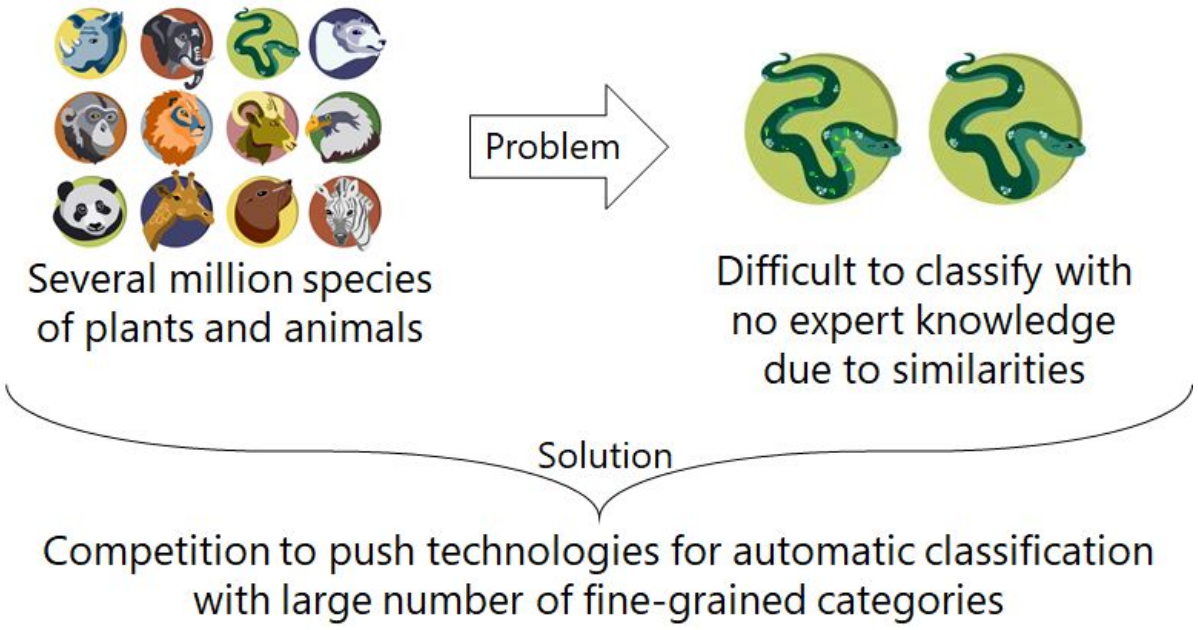
唐甫頌 Kim-Benjamin (0845058)
林彥岑 (0856125)
許可 (0840033)

Introduction

This is our report for the final project with the iNaturalist 2019 image classification task.

Motivation

The number of species of plants and animals are estimated to be around several millions and without expert knowledge, it is very difficult to classify the species due to their visual similarities. Therefore it is imperative to push the state of the art in automated image classification technologies. This would then allow to use real world data with a large number of fine-grained categories to train a model and let it automatically classify species correctly.
Compared to the previous Kaggle iNaturalist competitions, this one focuses on a much smaller but even more similar number of species.
The dataset contains 268243 images from 1010 different species, which have been collected and verified by multiple users from iNaturalist.



Challenges

The main challenges of our work are storage and memory. For the storage issue, there are over 85gb in .zip format, that was too much for Kaggle / Google Colab. For the memory issue, since our own computer had insufficient memory, e.g. 4GB ram, it is not enough to train and validate big models, such as Inception v3 and wide ResNet. As a solution, the dataset is stored in our local computer or server, and the training process of big models was executed in the lab server, with 12GB GPU.

Related Work

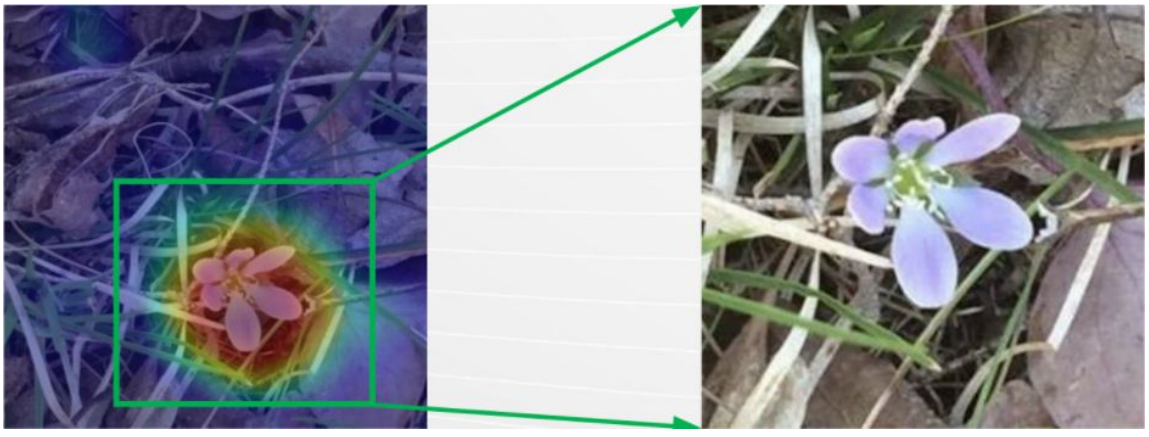
For the related work we looked at the different approaches by competitors that achieved high placements in the ranking (Top 5 out of 214 participants in the competition). A commonly used architecture with different modifications was ResNet-50, which was also used together in an ensemble of up to four models with other architectures.

Data processing

For data processing, besides data augmentation via scaling, flipping or adjusting the color of the images, weakly supervised localization with class activation maps (CAM) and complement entropy loss have been implemented. Moreover a high resolution of the images with dimensions for 560*560 pixels as well as class balanced sampling from the unbalanced data has been utilized as a countermeasure for overfitting towards one or more classes with more data points and an overall improved performance.

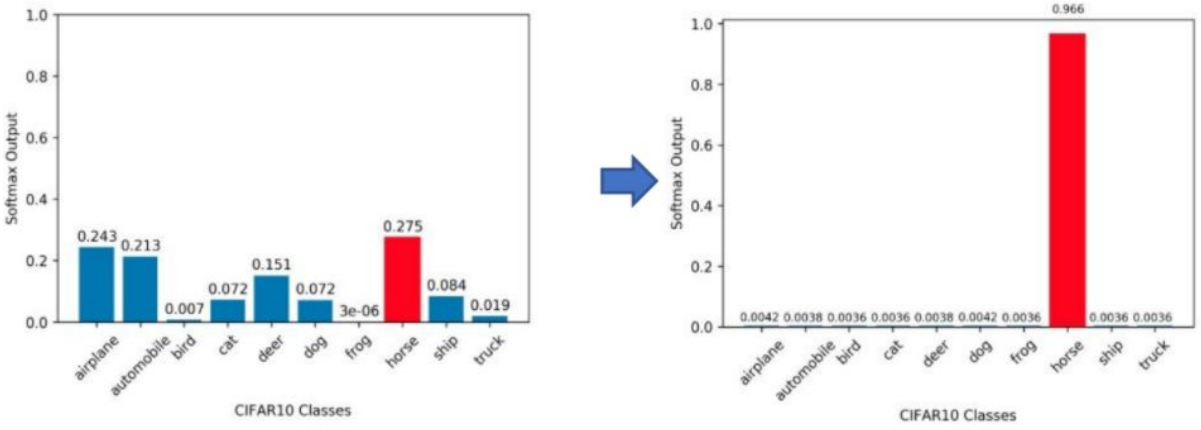
Supervised localization with class activation maps

For improved accuracy through small object localization supervised localization with class activation maps has been implemented. This means small objects in the input images were detected, so that the image could be suitably cropped to focus only on the relevant parts of the image. In the example image below it can be seen how the class activation map has been used to localize the flower. This allowed to focus on the flower in this image and helps to increase the classification accuracy.



Complement entropy loss

Another relevant technique is the complement entropy loss. By its implementation it was possible to maximize the likelihood of the ground truth while neutralizing the probabilities of complement classes.



(a) $\hat{\mathbf{y}}$ from the model trained with cross entropy.

(b) $\hat{\mathbf{y}}$ from the model trained with COT.

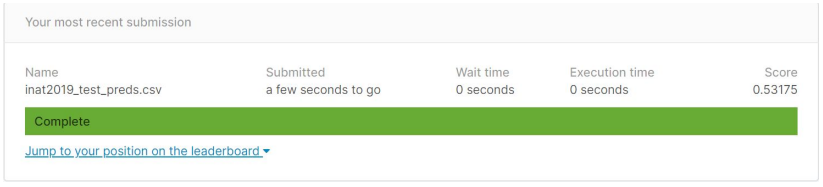
Proposed Approach

Owing to source restriction with the large dataset, we could only run models in our local device since both Google Colab and Kaggle had restrictions regarding the available storage space. The devices are equipped with 11GB and 4GB graphics memory, respectively. For wide ResNet and Inception v3, we ran our Code on the K40m for its huge consumption for graphics memory while we test MnasNet on the GTX 1050Ti for faster iteration.
This wide ResNet is a variant of ResNet-50-2, and its bottleneck number of channels is twice as large.
As for MnasNet, we try to adopt one with depth multiplier of 1.3. But we found out soon that the pretrained checkpoints are not available for that version. Therefore we used the version of 1.0 depth multiplier.

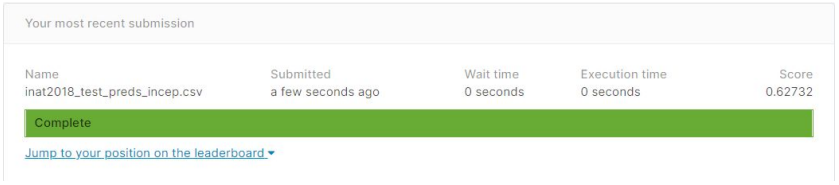
Experiment Results

In our code we tested different network architectures and experimented with different hyperparameters as well. The results from our experiments with varying epochs are shown below:

Score on MnasNet (18 epoch):



Score on Inception v3(6 epoch):



Score on wide ResNet (4 epoch):

Conclusion

Looking back at our work we can see that the data augmentation we implemented with flipping, scaling and also modifying the colors of the images have helped to increase the performance of our model. Although overall it is clear that it is a difficult task and takes a considerable amount of time and research to implement a high performing model. Moreover a lot of time and computational resources that aren't freely available on such a scale are required to perform hyperparameter fine tuning possibly via gridsearch for better training results.

References

- iNaturalist 2019 at FGVC6
- Wide Residual Networks
- MnasNet: Platform-Aware Neural Architecture Search for Mobile
- FGVC6 iNat Competitors - Public.pdf
- Images for animal icons
- Related work images