InceptionV3 Transfer Learning for Efficient Flower Classification / InceptionV3 modell használata hatékony virág kategorizáláshoz

Háhn Nikoletta Hedvig, Szommer Zsombor, Skáre Erik

***Abstract*—The project source is from Kaggle.com. The chosen task recommends training on TPU. However in this project we will use GPU. For this competition, our task was to create a machine learning model that recognizes different types of flowers in an image dataset. In this competition, we classify 104 types of flowers based on images from five different public datasets. Some classes contain many subtypes, such as wild rose, and some classes contain only one particular flower subtype. The goal of our project is for our model to recognize the flower from the images submitted. The image is submitted by the user through an API and the model classifies it using a probability distribution.**

***Abstract*—A projekt forrása a Kaggle.com. A kiválasztott feladat olyan feladat, amely a TPU-n történő tanítás tesztelésére szolgál. Ebben a projektben azonban GPU-t fogunk használni. A versenyen a feladatunk egy olyan gépi tanulási modell létrehozása volt, amely felismeri a különböző virágtípusokat egy képi adathalmazban. Ebben a versenyben 104 virágtípust osztályozunk öt különböző nyilvános adathalmazból származó képek alapján. Néhány osztály sok altípust tartalmaz, például vadrózsa, és néhány osztály csak egy adott virág altípust tartalmaz. Projektünk célja, hogy modellünk felismerje a virágot a beküldött képekből. A képet a felhasználó egy API-n keresztül küldi be, a modell pedig egy valószínűségi eloszlás segítségével osztályozza azt.**

***Index Terms*—Transfer learning, Neural networks, InceptionV3, Flower classification**

I. Introduction

In the realm of computer vision, the art of classifying flowers stands as a compelling challenge. Numerous technologies can be used, yet they vary in effectiveness. The goal is clear: to develop a model that not only recognises the array of flower species but also effectively adapts to variations in lighting, backgrounds. For this use case it is convinient to use CNNs[9] however with increasing performance requirements for training we needed to find a way to train an effective model with a reasonable time interval. This article embarks on a journey into the world of flower classification, wielding the power of transfer learning. Our approach involves using a pre-trained convolutional neural network (InceptionV3) and adding our own model on top of it which will be the only part trained.

II. Description of the subject area previous solutions

The subject area of our paper lies at the intersection of computer vision, machine learning. With the power of CNNs we seek to win out generalized knowledge and fine-tune the model to discern the unique features describing each individual flower species.

Various algorithms has been proposed to solve this task including pairwise rotation invariant co-occurrence local binary pattern[1], color attention-based bag-of-words approaches[2], etc. Recent times have proven that CNNs by far are the most effective. Most of these models take advantage of transfer learning because research shows that it can be powerful in computer vision[3]. Also it seems that data augmentation can increase performance[4] however due to resource constraints we did not include it in this paper.

III. Implementation

A. Data acquisition, preparation

We used the dataset for the **Petals to the Metal - Flower Classification on TPU** Kaggle competition. This was already split into training, test and validation datasets by default, so we didn't need to bother with that. The dataset contains 104 flower classes of different sizes. We chose 224*244 for training our model, as this offers both relatively good quality and reasonable training time[10].

B. System design

Taking into consideration the size of our dataset which is relatively small the decision was made, we must use the transfer learning technique. However there are a lot of pre-trained models for such purpose, by doing our research we concluded that InceptionV3 could provide a great accuracy[5][7]. We loaded the model without the top layer and added three hidden layers with ReLu activations and an output layer with softmax activation for receiving a probability distribution as a result. We also added dropout layers inbetween to prevent overfitting[8]. But the hyperparameter-space is gigantic, somehow we needed to reach the optimal configuration. That is where hyperparameter-optimization algorithms come into play. For this project we used Bayesian optimization technique[6] with early stopping. This led to a 72 percent accuracy on the validation dataset.

C. Training

We built and trained the model with the Keras framework. However somehow we needed to track the progress of the optimization and the final training. For these purposes we used

Weights and Biases. We created a sweep which contained all the information for the hyperparameter-optimization. Every iteration ran for maximum of 20 epochs and had early-stopping enabled after 4 epochs of unsuccess. After a few bad attempts the probability-based optimizer reached a point where it was able to generate models with 70 percent plus accuracy. This is where we stopped because the accuracy has stopped increasing. We took the best model's configuration and trained it for 100 epochs (also with early-stopping). It has reached our final accuracy which was 72 percent.

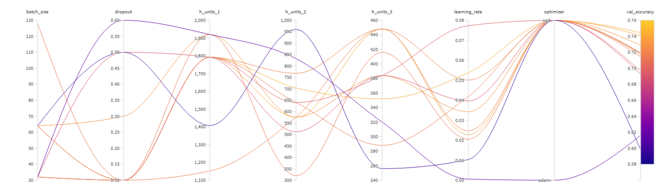

Fig. 1. Hyperoptimization in progress

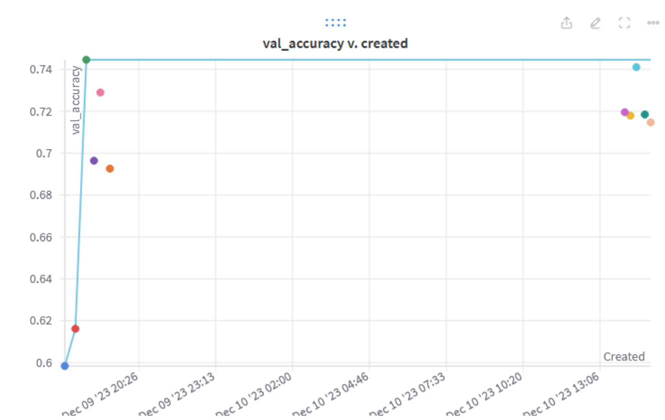

Fig. 2. Validation accuracies of each iteration

D. Gradio

Gradio is a Python library that allows to quickly create customizable UIs for simple machine learning models, thus we created a Gradio app for our model.

Hugging Face Spaces is a platform provided by Hugging Face that allows you to host and share machine learning models. We made the Gradio app available through Hugging Face Spaces.

IV. Evaluation

The figure one display the accuracy of the model, that shows early stopping in 35 epoch that highest accuracy 0.9935 of the training curve.

Figure 2 shows confusion matrix. F1 score is calculated for each class/labeled and then averaged.

V. Future Plans, Summary

Although the current phase of the flower classification project has come to an end, there are several possible directions for future work.

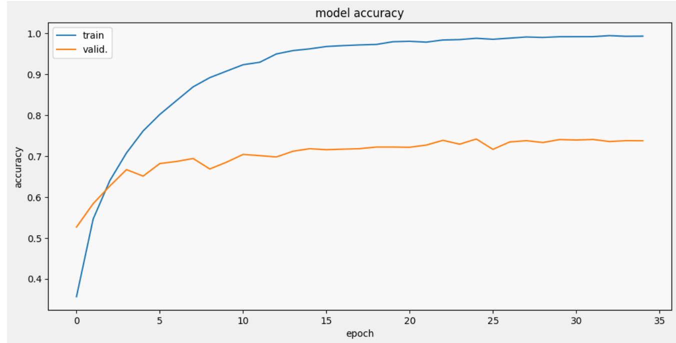

Fig. 3. Accuracy of the model

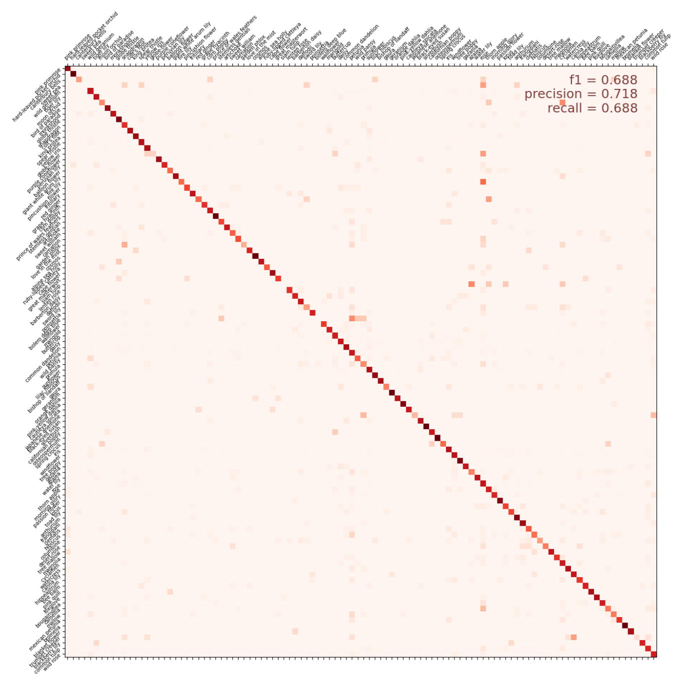

Fig. 4. Confusion matrix

First, the model could be extended to recognise more flower types. This would require collecting and labelling more data and possibly retraining or fine-tuning the model based on the new data.

In addition, the performance of the model could be further improved. Although the current accuracy is satisfactory, there is always room for improvement in machine learning projects. This may involve experimenting with different model architectures, tuning the hyperparameters more extensively, or applying more advanced techniques to augment the data.

In conclusion, although the current project is complete, the skills and knowledge gained in it will be valuable for future work on machine learning and image classification.

References

[1] *Pairwise Rotation Invariant Co-Occurrence Local Binary Pattern*, by Xianbiao Qi, Rong Xiao, Chun-Guang Li, Yu Qiao, Jun Guo and Xiaoou Tang

[2] *The Impact of Color on Bag-of-Words Based Object Recognition*, by David Augusto Rojas Vigo, Fahad Shahbaz Khan, Joost van de Weijer and Theo Gevers

[3] *A Decade Survey of Transfer Learning (2010–2020)*, by Shuteng Niu, Yongxin Liu, Jian Wang and Houbing Song
[4] *Improving Deep Learning with Generic Data Augmentation*, by Luke Taylor and Geoff Nitschke
[5] *Inception-v3 for flower classification*, by Xiaoling Xia, Cui Xu and Bing Nan
[6] *Hyperparameter Optimization for Machine Learning Models Based on Bayesian Optimization*
[7] *Robust Sports Image Classification Using InceptionV3 and Neural Networks*
[8] *Towards dropout training for convolutional neural networks*, by Haibing Wu and Xiaodong Gu
[9] *Flower classification via convolutional neural network*, by Yuanyuan Liu, Fan Tang, Dengwen Zhou, Yiping Meng and Weiming Dong
[10] *Effects of Varying Resolution on Performance of CNN based Image Classification: An Experimental Study*, by Suresh Prasad Kannojia and Gaurav Jaiswal