Mini Paper

Flower Power III

Large Scale Flower Monitoring

Veneta Angelova
Department of Software Engineering Fontys University of Applied Sciences Eindhoven, The Netherlands
v.angelova@student.fontys.nl

Kristina Krasteva
Department of Software Engineering Fontys University of Applied Sciences Eindhoven, The Netherlands
k.krasteva@student.fontys.nl

Lia Boyadzhieva
Department of Software Engineering Fontys University of Applied Sciences Eindhoven, The Netherlands
l.boyadzhieva@student.fontys.nl

Filip Vangelov
Department of Software Engineering Fontys University of Applied Sciences Eindhoven, The Netherlands
f.vangelov@student.fontys.nl

Abstract - Over the last few decades researchers have discovered significant decline in wildlife species and have also observed that some species are in danger of becoming extinct. Therefore, it has become of utmost importance to monitor wildlife closely. Hence, the goal of this project is to create a model which can recognize and count wildflower species. The applied approach is the IBM CRISP-DM model. The achieved results by the three models are quite promising. The custom and the pre-trained model are able to differentiate 30 flower species whereas the Faster R-CNN only 5 species. Each model can be applied according to the available resources which are needed for the performance of the models.

Keywords - **biodiversity, wildlife extinction, technology and urbanization, pollinators, ecosystem, large scale monitoring**

I.INTRODUCTION

With the rapid development of technology and urbanization over the last 50 years, the biological cycle has profoundly changed. Farming has become a leading industry in many countries, the Netherlands including. The excessive use of land for profit and the polluting of the soil with pesticides has undoubtedly influenced the biodiversity of the natural landscape. Furthermore, these activities alongside other factors, are causing the extinction of a large number of wildlife flower species. These species form on a large level the insect pollinator fauna, and its extinction is causing a significant effect on insect pollinators, who have a vital role in the ecosystem integrity and maintenance - they account for the production of nearly 85% of wildlife flower species. Furthermore, studies have discovered a decline of approximately 40% of all pollinator species globally during the past few decades. These global declines have highlighted the importance of wildlife flower species and has created the urge for large scale monitoring for their presumption. However, large scale monitoring requires extensive labor work and is prone to human errors, since it is performed manually. With the rapid development of technology and artificial intelligence, the IT sector has achieved promising results towards achieving this goal.

II.APPROACH

For this project, our team was provided with a dataset containing approximately 11GB of image data of flower species alongside an Excel sheet providing additional information regarding each image. Furthermore, our team was working closely together with Gerard Schouten(Professor AI & Big Data at Fontys University of Applied Sciences and head of the lectorate AI & Big Data.) towards the goal of the project. For the successful completion of this project, our team decided to adopt the approach suggested by the standard IBM CRISP-DM model, which consists of the following stages - Business understanding, Data understanding, Data preparation, Modeling, Evaluation and Deployment. During the initiation phase of our project our team was mainly focused on the business understanding of the project, as well as the data understanding. During the next stage, our team was mainly concentrating on the data preparation, and as the data was not labeled we devoted additional time on labeling and preprocessing the image data by using 2 different approaches, as the models which we used had different requirements regarding those aspects. Additionally, during the final stage of our project, our team was focused on modeling, evaluation and deployment. Furthermore, we decided to create 3 different models - Custom model, Faster R-CNN and Pre-trained Model (MobileNet_V2).

III.GOALS

The goal of the project is to monitor and differentiate between different types of flower species over large areas, which would have a great impact on biodiversity and ecosystem related complications. Provided that our project is successful, our model would give valuable insights such as areas with tremendously decreased amounts of wild flowers, insights regarding the multiplier effect on other species affected and other ecosystem related disturbances.

IV.EXPERIMENTS

Initially, our team and Gerard agreed upon developing a Faster R-CNN model. However, due to the extensive hardware which this model requires, we decided to create 2 additional models - Custom CNN model and Pre-trained Model (MobileNet_V2).

The implementation of the Faster R-CNN model required the use of LabelImg for the labeling of the images. However, during the testing of the model we discovered that images needed to be resized to 800x600 in order to have sufficient computing power to execute the model.

For our custom CNN model, we used our own automated labeling of the images and resized them to 128x128 pixels in order to have the same dimensions for all images. For building the model we used “Sequential()” from Keras, and consisted of 4 convolutional layers, 2 max pooling layers, 1 flatten layer and 2 dense where the last dense layer uses “softmax” activation. However, in the process of adding more species to be trained we customized the parameters, added one more dense layer and a dropout of 50%. For compiling, the ”Adam” optimizer was applied with a learning rate of 0.0001.

The third model which we used was MobileNet_V2, which is a classification model developed by Google. Furthermore, this model makes use of the feature vector, which aims to give all the layers of the pre-trained model except the last one which is determined by us. The “trainable” property has to be set to false which means to freeze and not train (all the layers will have their fixed weights).

V.RESULTS

Our group manages to achieve quite promising results with all three models.

The Faster R-CNN model was successfully trained for 15,000 steps on 5 different species by using the Faster R-CNN Inception ResNet V2 1024x1024 model. Furthermore, we noticed a steady decrease in loss over time, with an average of 0.05 - 0.1 for classification loss and 0.12-0.2 for localization loss.

The custom CNN model managed to reach a validation accuracy of 76% for 30 flower species. Despite that we evaluated the model by plotting the validation accuracy and loss as well as a confusion matrix, we plot an evaluation on the dataset. This is done by displaying an image from the dataset with the true label above and the predicted label by the model.

The pre-trained model achieved an accuracy score of 98% for the 30 species it was trained with. The same approach as the custom model was used to visually show and evaluate the performance of the model on the test set. Therefore, a random subset of 36 images of the whole test set was used for its evaluation.

VI.CONCLUSION AND RECOMMENDATIONS

Our research concluded with quite promising results from all 3 models. The accuracy scores from both the custom CNN and the pre-trained model (MobileNet_V2) are very promising, as well as the performance of the Faster R-CNN. However, for experimenting further with the Faster R-CNN model our group would strongly advise towards the use of a more powerful hardware as

the model requires a great amount of computing power.
Furthermore, we strongly encourage further experimentation with parameter optimization regarding all three models, as well as introducing additional images and flower species, which would be an interesting continuation to the work already done. In this way it could be determined whether or not the pre-trained and the custom models are actually performing well. Furthermore, it would also be interesting to experiment with other pre-trained models and determine if even better results can be achieved.

REFERENCES

1.Following the data science methodology. (n.d.). IBM Developer. Retrieved September 30, 2021 from https://developer.ibm.com/blogs/following-the-data-science-methodology/

2.Step-by-step data mining guide. (2000). Chapman, P., Clinton, J., Kerber, R., Khabaza, T., Reinartz, T., Shearer, C., & Wirth, R. Retrieved September 30, 2021 from https://the-modeling-agency.com/crisp-dm.pdf

3.Gandhi, R. (2018, July 9). R-CNN, Fast R-CNN, Faster R-CNN, YOLO — Object Detection Algorithms. Towards Data Science; Towards Data Science. Retrieved October 25, 2021 from https://towardsdatascience.com/r-cnn-fast-r-cnn-faster-r-cnn-yolo-object-detection-algorithms-36d53571365e

4.Ren, S., He, K., Girshick, R., & Sun, J. (2017). Faster R-CNN: Towards Real-Time Object Detection with Region Proposal Networks. IEEE Transactions on Pattern Analysis and Machine Intelligence, 39(6), 1137–1149. Retrieved October 25, 2021 https://proceedings.neurips.cc/paper/2015/file/14bfa6bb14875e45bba028a21ed38046-Paper.pdf

5.Training Custom Object Detector — TensorFlow Object Detection API tutorial documentation. (2017). Readthedocs.io. Retrieved November 14, 2021 from https://tensorflow-object-detection-api-tutorial.readthedocs.io/en/latest/training.html

6.Installation — TensorFlow Object Detection API tutorial documentation. (2019). Readthedocs.io. Retrieved November 14, 2021 from https://tensorflow-object-detection-api-tutorial.readthedocs.io/en/latest/install.html

7.Patel, I., & Patel, S. (n.d.). An Optimized Deep Learning Model For Flower Classification Using NAS-FPN And Faster R- CNN. Retrieved January 12, 2022, from http://www.ijstr.org/final-print/mar2020/An-Optimized-Deep- Learning-Model-For-Flower-Classification-Using-Nas-fpn-And-Faster-R-cnn.pdf

8.Why Google's MobileNetV2 Is A Revolutionary Next Gen On-Device Computer Vision Network. (2018, April 18). Analytics India Magazine. Retrieved January 12, 2022 from https://analyticsindiamag.com/why-googles-mobilenetv2-is-a-revolutionary-next-gen-on-device-computer-vision-network/

9.Kharwal, A. (n.d.). Flower Recognition with Python. Data Science | Machine Learning | Python | C++ | Coding | Programming | JavaScript. Retrieved January 12, 2022 from https://thecleverprogrammer.com/2020/11/24/flower-recognition-with-python/