21.6.2024

Plant Recognition

Final report

Inhalt [tribute to Germany 😊]

Abstract

This is the final report for the project **Plant recognition** that was carried out during the training course **apr24_bootcamp_ds** provided by DataScientest. Main purpose of this project was to allow the students to apply and test the methods learned during the training on a "real-life project".

The task in this specific project was to locate and classify the species of a plant in an image. Also, in scope was to detect potential diseases of that plant. Means the solution is expected to return this information to the user from an image taken by a camera.

Different datasets of suitable images publicly available in the Internet were identified by the project team, analyzed and structured. Then several machine learning and deep learning methods were implemented and tested. One of the deep learning models was selected to be the "winner model" as it was outperforming the others by reaching an impressive F1-Score of 97% for the test dataset. The team then selected, implemented and applied contemporary interpretability techniques on this model to better understand and to visualize how it is working in detail.

It's also worth to mention that conducting a project like this is coming with obstacles and challenges. A summary of those is included in this document, also to give the opportunity for future similar projects to learn from them.

To summarize, the team pursued highly motivated and passioned on implementing this project and reaching its objectives. Deep learning algorithms are known to be a typical use case for image detection problems. This was (once again) confirmed during this project. The potential to apply this technology on real life problems is enormous. We will see an outlook on this in the subject of plant recognition in the last chapter of this document.

1. Data Analytics

1.1. Dataset

The data used in this project has been retrieved from the following sources:

- The *Plant Seedlings* Dataset

This dataset contains images of very young plants (seedlings). It has around 5.5K images that represent 12 classes.

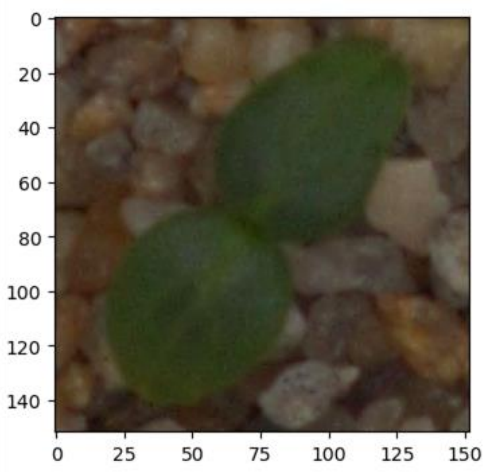

Figure 1: Representative image of the dataset [seedling of the plant species "Cleavers"]

- The ***PlantVillage*** Dataset

This dataset contains images of the leaves for grown-up plants. It has around 55.5K images. The dataset contains healthy plants but also plants with diseases.

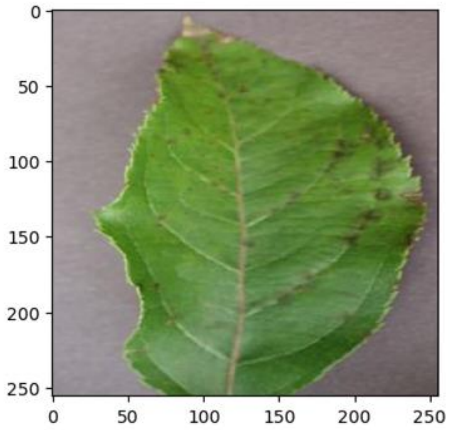

Figure 2: Representative image of the dataset ["scab"-diseased mature leaf of the plant species "Apple"]

So, the full dataset contains ~61K pictures. This consolidated dataset represents 26 different plants and shows 20 different types of diseases, in total 51 classes.

For more detailed evaluation we'd like to refer the reader to our project report 1 which contains in-depth analytics of the datasets used.

1.2. Evaluation criteria

We decided to use the F1-Score as primary evaluation criteria for any of our models. Important hereby is that this score has to be calculated on a test dataset which is to be kept separate from any of the training data for the model.

Additional criteria were that there are no obvious (or at least significantly worse than others) signs of overfitting. Also, the IT resources and time required to train the model were considered in the evaluation.

2. Model Selection and Justification

2.1 Comparison of Different Deep Learning Models

In our project aimed at developing a sophisticated plant recognition system, we carried out an in-depth evaluation of several deep learning models. This comprehensive analysis was vital to ensure that the chosen model would not only deliver high accuracy in identifying various plant species and detecting diseases but also fit our operational requirements for efficiency and deployment capabilities.

I. Initial Setup with Convolutional Neural Networks (CNNs)

Our exploration began by establishing an initial model using a traditional Convolutional Neural Network (CNN). CNNs are well-suited for processing pixel data and are commonly used for their robust performance in image classification tasks. Our initial CNN architecture incorporated two convolutional layers with ReLU activation functions, pooling layers to reduce spatial dimensions, and dense layers culminating in a Softmax layer designed for multi-class classification.

This initial model served multiple purposes:

- **Benchmarking**: It provided a foundational accuracy benchmark, achieving approximately 85% accuracy on our validation set. This figure was crucial as it set a baseline for future models.
- **Feature Learning**: The initial model allowed us to understand the basic features it could extract from the plant images, such as textures and colour patterns.
- **Model Limitations**: The most significant insight, however, was recognizing the model's limitations. It was evident that the model exhibited signs of overfitting and demonstrated relatively poor performance on some underrepresented classes. This observation led us to explore more robust solutions.

II. Exploring Advanced Architectures: LeNet and Transfer Learning

Given the shortcomings of our initial CNN model, we expanded our exploration to revisit and refine the LeNet architecture, which had formed the basis of our initial experiments. LeNet, historically significant for its role in advancing image processing, was initially applied to a subset of our dataset. This application included images of apples, strawberries, and sugar beets, focusing solely on the model's ability to classify these plant types.

Surprisingly, the LeNet model performed exceptionally well on this subset, achieving an F1-score of 95% on the test set without any hyperparameter tuning. This success prompted us to consider scaling the LeNet model to handle our entire dataset.

III. Scaling Up with LeNet

The process of scaling the LeNet model to accommodate our complete dataset of over 60,000 images across 51 classes involved overcoming significant challenges such as overfitting and class imbalance. To address these issues, we implemented several strategies:

- **Data Augmentation**: Introduced to combat overfitting and enhance the model's generalization capabilities.
- **Hyperparameter Tuning**: Adjustments were made to the learning rate, and the model's architecture was tweaked to better handle the diversity and complexity of the dataset.

IV. Transfer Learning with MobileNetV2

Concurrently with refining the LeNet model, we explored using transfer learning to leverage pre-trained models. Our focus was on MobileNetV2, initially trained on the comprehensive ImageNet dataset. MobileNetV2 was particularly appealing due to its computational efficiency and adaptability, crucial for mobile and low-resource environments.

MobileNetV2 excelled in several aspects:

- **Computational Efficiency**: Designed for efficiency, it proved capable of running on devices with limited computational power.
- **High Accuracy**: It demonstrated remarkable accuracy when fine-tuned on our dataset, effectively recognizing subtle differences between plant species and conditions.
- **Flexibility**: The architecture facilitated rapid training and easy adjustments.

2.2 Final Model Selection

After extensive testing and analysis, MobileNetV2 was selected as the final model for our plant recognition system. This decision was influenced by the model's impressive balance between accuracy and efficiency, robustness across diverse deployment scenarios, and scalability.

Notably, MobileNetV2 achieved over 97% accuracy on the validation set, highlighting its effectiveness. However, it is essential to note that while these results are promising, they do not necessarily guarantee

similar performance in real-world scenarios. Our limited testing with real-world data suggests that further validation is required before deployment. Thus, any potential future deployment should be approached with careful consideration and further testing.

In conclusion, the choice of MobileNetV2 aligns perfectly with our project goals, blending cutting-edge technology with practical implementation considerations. As we progress, our focus will remain on continuously optimizing this model through learning and adaptation, ensuring it is prepared for potential future deployment scenarios. This strategic choice not only fulfills our current technical and operational needs but also sets a robust foundation for future enhancements and expansions of our plant recognition capabilities.

2.3. Model Interpretability

I. Greeting approaches

The final stage of scrutinizing the performance of our winner model[1] was to carry out the interpretability check that raises the lid of its black box (which is worth to open, since even though fine-tuned by us, it essentially belongs to Google and not Pandora 😊) and lets us put our curious hands on some of its nuts and bolts.

Alright, enough talking figuratively, let's get straight to the gist of the matter: we gathered all our courage and skills to trespass the rough terrain of still largely evolving CNN interpretability research field, and despite lots of sweating blood and tears when:

- tackling the sparsely available code to make it work for our models, as well as
- taking laborious measures to ensure the theory/practice ratio of the thoroughly overhauled code is straightened out,

we succeeded in overcoming all these difficulties and implemented, to the best of our knowledge, the following couple of most informative of all the present-day C/NN interpretability techniques which, to put in a nutshell, grant anyone with the access to a compiled & trained (in our case, Keras) model the ability to visualize which parts of an input image and at which color-coded extent (class discriminatively) contribute to the outcome:

- **Grad-CAM** [1] that was introduced back in 2016, and since that went viral in the Data Science (Deep Learning) community amounting to tens of thousands of citations in the academia alone, and
- **Guided Grad-CAM** that, to put it roughly, represents a fine-grained version of Grad-CAM thanks to the basis of guided backpropagation intertwined with the Grad-CAM's heatmaps (for readers thirsty for more technical details and not allergic to some differential calculus, we refer to [2] for more details).

II. Grasping approaches

Before delving into a closer examination of interpretability outcomes or, in better words, of image-wise visual explanations for the model predictions, let's sneak a peek at our bride before the actual wedding in order to get a quick illustrative overview of all the analysis targets:

[1] Descriptively named „TL_180px_32b_20e_model.keras", nicknamed "BeLu" aka "champ" 😊, representing the fine-tuned Transfer Learning model on the basis of Google's MobileNetV2

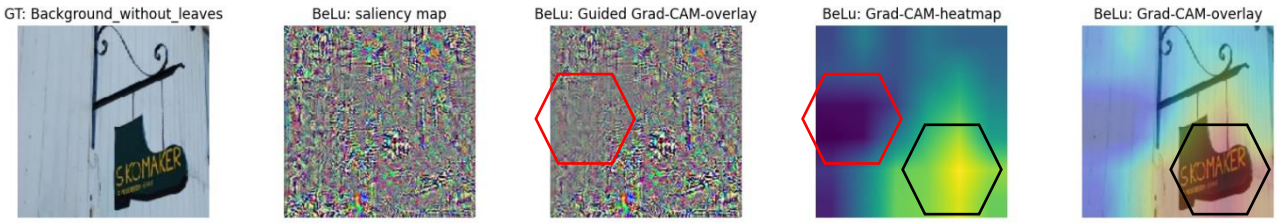

Figure 3: demonstrates the input image (left) together with its saliency map based Guided-Grad-CAM (center) as well as its heatmap-based Grad-CAM (rightmost)

The hexagonal region, marked in red, shows the least activated area of the image i.e. the one that contributed to the class score (in this case of the "Background_without_leaves" label) the least, whereas the black hexagonal area – the one that contributed the most. Saliency map does not exhibit this clear segmentation due to its nature of not being class-discriminative – a careful comparison of the saliency map vs Guided Grad-CAM from the closeup perspective could easily lead to the actually correct finding of attentive readers who reckons that the Guided Grad-CAM is the result of the overlay of its next-door neighbours. That's right! The same could be intuitively said about the Grad-CAM overlay that results from blending the input image with the Grad-CAM heatmap.

In case a scrupulous reader objects that there is no need in having saliency maps and thus also Guided Grad-CAM, a good soothing and simultaneously sobering reply would be to point out its great advantage over the heatmap i.e. generally way higher resolution, in fact, as high as of the input image, whereas the heatmap, say, in this particular case, is extrapolated from tiny 6x6 pixels (due to MobileNetV2's intrinsic design that comes armed with such last non-1x1 convolutional layer neurons – but no worries, there are 1280 of them) in order to match the 180x180 px size of the input image. So, as Donald would say, it's huge!

III. Employing approaches

Without further ado and platitudes, let's get to the point:

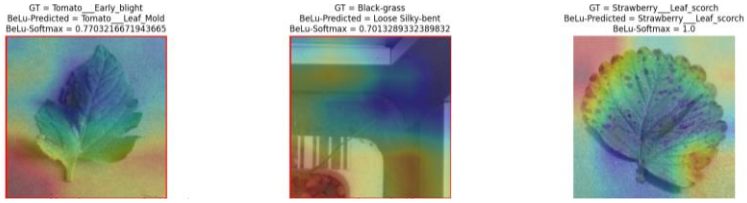

Figure 4: shows the threesome from randomly selected sample of test & validation images of our raw dataset, red-framed are false predictions, GT stands for the "ground truth" class

The upper subsample represents the trio of classification outcomes overlayed with the Grad-CAM heatmap, which provides nearly perfectly clear and reasonable interpretation to the actual outcomes presented in the titles: the only correct prediction, represented by the rightmost image, comes with the highest Softmax-score that could be interpreted as the probability of belonging to the, in fact, true class. And the overlayed Grad-CAM heatmap shows upper-left and both lower- and upper-right parts of strawberry leaf edge zones as hot regions which, most importantly, also contain the spots characteristic for the disease named "leaf scorch". A sceptical reader might notice that, along with hot spotted areas mentioned above that surely contributed to the observed true prediction, the hot spots also embrace background fragments, and, surely, these artifacts could be easily interpreted as apparent signs of (a) model overfitting relative to the truly unseen data, and (b) as evident indication of need in more complex treatment/preprocessing of images, say, via segmentation. But, again, Grad-CAM showcases some vividly clear interpretability of our winner model's outcomes, irregardless of being positive or negative.

Observing the first two images (from the left) brings us again to the outstandingly logical interpretations of mispredictions as well in both cases being the hot spotted background areas with little to no intersection with the actual foreground of classification i.e. leaves/plant parts. Moreover, the Grad-CAMs allow us to dig even deeper and, for instance, additionally notice that the image in the center was generally utterly hard to predict since it depicts a seedling plant of Black-grass that covers only a miniscule part of the extremely heterogeneous image area, whereas for the leftmost image our winner model mispredicted the disease, but still correctly identified the plant species, apparently due to the fact that hot spots still covered lower contours of the tomato leaf. In fact, we could deduce even more: the Black-grass class is one of the very few underrepresented classes in our dataset, and since this image could be taken as a class representative (due to the rather disadvantageously homogeneous constitution of our dataset in general and of this class in particular), we could state with solid confidence that nothing but adding more closeup images or zoom-in augmented derivatives has chances of boosting the hit rate for this particular class.

Also the Guided Grad-CAM turns out to be more than helpful in interpreting handful of (mis)behaviors:

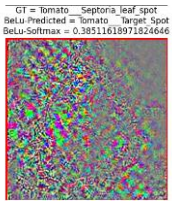

Figure 5: Guided Grad-CAM of model' false prediction example

Even though failed in discerning spots of different disease, the Guided Grad-CAM due to its pixel-wise precision subtly showcased that the winner model focused, in fact, on the entire characteristic contour of the tomato leaf, resulting in true prediction of species.

Yet another fine-grained interpretation of the model behaviour on this particular image – again, the winner model succeeded practically in precise segmentation of the most part of the target plant leaf, leaving no chances of mishitting a shot.

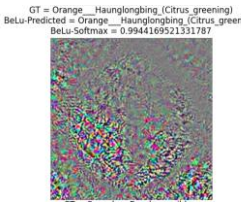

Figure 6: Guided Grad-CAM of winner model's true prediction

All in all, after the interpretability analysis, the inner core of our winner model i.e. the fine-tuned MobileNetV2 became less of an enigmatic black box, and gave us no reasonable evidence to dethrone it, even though we had some decent competitors in our ammunition storage like the fine-tuned LeNet-family members of our own production, both with and without augmentation layers.

3. Difficulties Encountered During the Project

3.1. Scientific Obstacles

Throughout the project, our team encountered several scientific challenges that significantly impacted our progress and required innovative solutions. Some of these challenges are:

- None of the team members had domain knowledge of plant diseases, so we relied on external sources for the categorization of plant diseases as healthy or non-healthy.
- Integrating diverse datasets from multiple sources with varying image quality, resolution and semantic discrepancies (datasets were collected for different purposes) added another layer of complexity.
- Since each species has unique characteristics, such as leaf shape, colour, and texture, which can vary significantly and also varied plant diseases can manifest in many ways, including spots, discoloration, and wilting. These symptoms can look different depending on the plant species and the severity of the disease. Ensuring the model could correctly identify the same disease across different plants (some plants share the same disease) and disease stages (even the same

disease (some of them) in the same plant look different at different stages) was challenging. Ensuring the robustness and generalizability of our model across different plant species and disease types was a constant scientific challenge.

- Each team member came from a different background—mathematics, management, software development—so the obstacles varied for each person.

3.2. Challenges

I. Timeline Management

There were some other delays due to unforeseen issues, which we managed to compensate for by utilizing time more efficiently on less challenging tasks. Some specific delays included:

- **Model Training Time and Fine-Tuning:** Training deep learning models, especially with limited computational resources, was time-consuming.
- **Coordination Issues:** Remote work sometimes led to miscommunication and delays in task completion.

These challenges are typical in projects involving multiple team members and complex tasks.

II. Datasets

Selecting the ideal dataset presented a significant challenge due to concerns over the large formatting and semantic discrepancies between the datasets listed in the Project Description file. Although both the Seedlings and PlantVillage datasets were identified at the project's inception, integrating them posed several difficulties. The Seedlings dataset consists of whole plants photographed in styrofoam boxes, where none are diseased, while the PlantVillage dataset includes images of harvested leaves from grown plants, featuring both healthy and diseased specimens.

Our initial brainstorming sessions focused on how to handle these discrepancies and the specific treatment of the Background_without_leaves (a class with random places and objects) class. This particular class introduced noise and randomness into the dataset. Including such a class is crucial for:

- Model Robustness: Helps the model distinguish between relevant and irrelevant features, reducing false positives.
- Generalization: Improves the model's ability to handle new, unseen environments by exposing it to diverse backgrounds.
- Edge Case Handling: Prepares the model for real-world scenarios where plants/crops (leaves) might be obscured or partially visible.

We debated whether to classify the species first and then determine the health status or to identify them directly in the "as it is" format provided. Ultimately, we chose the latter approach, which involved combining the Seedlings dataset with the PlantVillage dataset, adding 12 more classes to our existing ones. By the end, we had a comprehensive dataset with 51 classes of plants and their diseases, significantly improving our model's robustness.

III. Technical/Theoretical Skills

Our project required acquiring new skills in advanced deep learning techniques not covered in our initial training. Specifically:

- **Deep Learning:** We had to delve deeper into convolutional neural networks (CNNs) and their architectures, learning how to design and implement models suited for image classification tasks.

- **Transfer Learning:** Implementing transfer learning using pre-trained models required a significant amount of work to learn the ropes of dealing with Keras API and how to fine-tune them for our specific task.
- **Hyperparameter Tuning:** Learning and applying techniques to optimize hyperparameters was crucial to improve model performance but required substantial experimentation and time.

IV. Relevance

Ensuring that our approach and model were relevant to the specific problem of plant disease detection posed another challenge. Given the pedagogical nature of the project and our team's lack of prior experience, we had to balance learning state-of-the-art deep learning techniques with practical limitations. Our time was restricted, and we relied on limited computing power, such as limited free tiers of Colab's GPU and our own standard laptops.

While we aimed to develop a practical model for real-world agricultural settings, these constraints left us unable to undertake measures to get our models truly deployment-ready. Despite our extensive literature review and testing, the performance of our models on real-world data was limited due to mentioned constraints. For example, the best-performing model correctly identified only 3 out of 10 images taken from the internet (images not similar to our dataset but from the same plants that we had in our classes).

V. IT

Limited Computational Power was IT-related challenge that we faced during the project. Training deep learning models is resource-intensive. Our primary computational resource was a standard laptop with limited GPU capabilities (only one team member had this) and the free tiers of Google Colab, which significantly slowed down the training process.

4. Achievement of Project Goals

4.1. Analysis of Goal Fulfillment

According to the goals outlined in our previous reports, the project can be considered successful based on several key metrics:

- **Accuracy:** Our final model achieved an impressive accuracy of 97% on the validation/test set.
- **Model Interpretability:** To examine the interpretability of our model, we employed both Grad-CAM (Gradient-weighted Class Activation Mapping) and its fine-grained competitor Guided Grad-CAM. These techniques visualize which parts of the input image were most influential in the model's decision-making process. This not only aids in understanding and trust-building among end-users, particularly farmers and agricultural experts, but also assists in diagnosing potential issues with model predictions. By highlighting the regions in the images that the model focused on, (Guided) Grad-CAM provides valuable insights into how the model differentiates between healthy and diseased plants.

4.2. Potential Applications of the Model

Our model, while not yet deployment-ready, demonstrates several potential applications that could significantly impact the field of agriculture if further developed and refined:

I. IoT Devices

The model can be integrated into IoT devices for real-time plant disease detection. These devices could continuously monitor crops, providing immediate alerts when a disease is detected. This rapid response

capability would allow for quick intervention, potentially saving entire harvests and improving overall crop management.

II. Agriculture Settings

In agricultural settings, the model can be used by farmers to monitor the health of their crops. By utilizing smartphones or dedicated camera systems, farmers could quickly identify diseases and take appropriate actions. This application would empower farmers with timely information, helping to maintain crop health and productivity.

III. Real-time Monitoring

For large-scale agricultural operations, the model could be part of an automated monitoring system. Using drones or stationary cameras, the system could scan fields and provide real-time updates and alerts about crop health. This would enable large-scale operations to manage extensive areas efficiently, ensuring that any disease outbreaks are promptly addressed.

In conclusion, despite the challenges encountered and the current limitations of our model, our team successfully developed a robust plant disease detection framework. With further development, access to heterogeneous datasets, and improved computational resources, this model has the potential to revolutionize agricultural practices. By enhancing the accuracy and applicability of AI in agriculture, we can significantly improve agricultural productivity and sustainability.

Closing

During the project we explored and applied machine and deep learning on an image classification problem with great success! The performance of the model with reaching an F1-Score of 97% on the test dataset is really impressive. However, it's also important to mention the limitations reached and to reflect on how those could find input into a future project.

As we have seen - the results for some of the classes with low representation significantly deviate from this average. Another important finding is that the model benefits from having a very homogeneous dataset: We also tested our model with images randomly selected from the internet (and thus not being part of the original dataset). With those only 3 out of 10 images got detected correctly. Now this small sample size obviously is non-representative, but significantly impaired performance on truly unseen and differently constituted images was well-expected.

So how could these findings be addressed?

First obvious thought is to apply more training for the classes where the model shows low performance. This could be achieved bluntly by obtaining more training data. However, there are limitations to obtain training data of good quality. Therefore, instead image augmentation could be used which is a general approach to address limitations from having small training sets. It is a technique that is tweaking existing data and those generating more training data. We explored and tested this approach during the project. Interestingly we weren't able to get better predictions for the minority classes by this and that's also why we didn't pursue to include this approach into our final model. However worth to mention is that another typical benefit of augmentation is to address overfitting. The positive effect of augmentation on preventing overfitting was clearly becoming visible during our trials. So, this topic potentially could be explored further by putting more resources into it.

We had the opportunity to apply different algorithms in practice during this project. However, the number of algorithms, their potential combinations and tuning the related hyperparameters practically is endless. We are very happy with the results obtained in the given timeframe, but our expectation is that even better solutions could be obtained through additional research or future technology. The most promising construction site would be to integrate both semantic and instance segmentation

approaches which possess all the potential to crank up the performance of, in fact, all our models up to raising them on a qualitatively new level.

As we conclude this project, it's essential to reflect on our journey. We encountered challenges and celebrated our achievements. We hereby like to thank the team of DataScientest and our project mentor Romain for allowing us to gain this valuable experience

References

1. Selvaraju, R. R., Cogswell, M., Das, A., Vedantam, R., Parikh, D., & Batra, D. (2017). Grad-cam: Visual explanations from deep networks via gradient-based localization. In Proceedings of the IEEE international conference on computer vision (pp. 618-626).

2. Molnar, C. (2022). A guide for making black box models explainable. https://christophm.github.io/interpretable-ml-book/index.html Accessed on 21.06.2024