11.5.2024

Plant Recognition

Data Science Project Scope

Inhalt [tribute to Germany ☺]

Team: Luigi Menale | Bernd Brinkmann | Arif Haidari | Alex Tavkhelidze

mentor: Romain Lesieur

DATASCIENTEST

Data Science Project Scope

Since the present data science project is predominantly, if not exclusively, framed to serve pedagogical goals, it becomes abundantly clear to identify the key stakeholders:

1. in the first place, our four-member team consisting of:

-lm- Luigi Menale, experienced Software Engineer

-bb- Bernd Brinkmann, experienced computer scientist in supply chain management

-ah- Arif Haidari, Software Developer

-at- Alex Tavkhelidze, your humble servant *(just a metaphor, ain't processing any orders :D)* mathematician

2. Romain Lesieur, experienced project mentor, representing ***DataScientest*** and hosting the project repository on GitHub (world's largest open-source code host, parented by Microsoft and based on the distributed version control system Git).

3. in the second place, any physical entity, interested in object recognition in general, and in its application in botany in particular. The interest might constitute the usage of publicly (e.g. over GitHub) available solutions to identify plant species and/or plant diseases on images, fed to the developed application, as well as employing Python libraries in order to enhance or develop similar tools. A proper attribution of authorship is advised.

1. Understanding the problem

In the light of emerging world population as well as impairing effects of global warming on crop yields, any development of innovative (AI-) solutions, which are practically oriented, very affordable and aid in spotting of spreading diseases, is more than welcome. Even on the level of a single farmer, a gardener, even an average Joe or a plain Jane who grows beloved plants in pots, it is handy and at the same time vital to be able to give an affected plant region a camera shot and get some immediate and helpful guidance for further steps. Thanks to the well-globalized technological markets and well-connected satellite internet *(shout out to Elon)*, a mere Web-connected smartphone equipped with a decent camera becomes an increasingly common thing in everman's hands. Such a world simply begs for reliably accurate and yet simple cloud-based AI-solutions that are able to robustly identify (at least major if not ideally any) creeping dangers for plants, which undoubtedly represent the primary link of the food chain, in one or few camera shots aka display touches.

Since it is apparently beyond our reach (as well as out of pedagogical focus of the present project) to serve the needs of remote/airborne sensing that covers large areas of vegetation quickly and since highly multispectral up to hyperspectral cameras/scanners is just a luxurious rarity for an average interested party, we reduce the target domain to RGB-channeled file formats, *.png* and *.jp(e)g*, which are the most common ways to store digital images and at the same time widely supported even by affordable digital cameras.

Even though you can easily come across some decent solutions that identify plant species via taken images (provided by ***PlantNet***, ***PictureThis***, ***Plant App***, ***Blossom*** as smartphone applications, just to name few, but well-reviewed by the general public), and most of them additionally identify diseases as well, the only decent app that offers the identification services unlimitedly free of charge, ***PlantNet***, has no feature of detecting diseases as well. With the present data science project, we'll try to make our first and best step to contribute towards filling this gap.

And last but not least, talking about identification of plant and diseases, we feel the urgent need to stress the following limitations, driven by the shortage of publicly available data as well as by the narrow scope and time margins of the current project:

- plant species identification covers only few classes of seedlings (focus area: entire aboveground part) and grown plants (focus area: mature leaves)

- only a narrow range of diseases of grown plants is considered (focus area: mature leaves)

- no further details like identification of further plant features rather than just species/diseases or generating advices (e.g. on care plans or preventing/combating diseases) are put on the current agenda

2. Defining the goals

The objectives of this project is to develop the application, written in the most ML-adapted and currently TIOBE-topping general-purpose programming language Python, that:

-0- allows a user to load an input which has to be a 2D digital image, taken by a camera

-1- locates plants in the input

-2- for each located plant:

-2-1- classifies the species of a plant

-2-2- identifies diseases of a plant

-3- returns the output to the user, containing for each located plant:

-3-1- its location on the input image

-3-2- description of classified plant species

-3-3- description of identified disease(s)

Possible versatility implementations with regard to the objectives listed above:

-0- possibility to load packs of images at once:

-0-1- files of archive format

-0-2- directories of files

-3- returning the input image(s) with [interactively] (annotated/labeled) bounding boxes of identified/detected plants

3. Defining what data do you have and what data do you need

I. dataset "species" for classes of plant seedlings species:

-o- official name and source: The Plant Seedlings Dataset

-i- total size of the dataset (without segmentation): ~1,59 GB (1.714.293.106 Bytes)

-ii- total amount of classes: #12

-iii- total amount of files: #5539

-iv- files per class on average (rounded up): #462

-v- typical file format description: PNG image data, 241 x 241, 8-bit/color RGB

-vi- short description of the file format: PNG is a raster-graphics [two-dimensional picture as a rectangular matrix or grid of pixels] file format, is compressed, i.e. needs en/decoding, in lossless way

-vii- aspect ratio: 1:1 (mostly square images; some of them are almost square with aspect ration ~1.0x)

-viii- image px (pixel) resolution: ~49x49 up to ~3129x3129 (~410 x 410 px on average)

-ix- classes/labels:

PSC01. Maize

PSC02. Common wheat

PSC03. Sugar beet

PSC04. Scentless Mayweed

PSC05. Common Chickweed

PSC06. Shepherd's Purse

PSC07. Cleavers

PSC08. Charlock

PSC09. Fat Hen

PSC10. Small-flowered Cranesbill

PSC11. Black-grass

PSC21. Loose Silky-bent

-x- short evaluation of the image contents:

---1- mostly heterogeneous background - brownish granular texture

---2- foreground represents entire seedlings (embryonic shoots & seed leaves)

---3- variance in image resolutions is rather large

---4- no common plant species with the dataset II (i.e. "species-n-diseases")

II. dataset "species-n-diseases" for classes of grown plant species and diseases:

-o- official name and source: The ***PlantVillage*** Dataset

-i- total size of the dataset (without augmentation): ~821 MB (861.528.492 Bytes)

-ii- total amount of classes: #39 (#38 - actual plant-n-health classes, #1 - in/out-door background images "Background_without_leaves")

-iii- total amount of files: #55448

-iv- files per class on average (rounded up): #1422

-v- typical file format description: JPEG image data, JFIF standard 1.01, precision 8, 256x256, components 3

-vi- short description of the file format: PNG is a raster-graphics [two-dimensional picture as a rectangular matrix or grid of pixels] file format, is compressed, i.e. needs en/decoding, with losses

-vii- aspect ratio: 1:1 (mostly square images; #1 class of background images "Background_without_leaves": uncommon 4:3 aspect ratio)

-viii- image px (pixel) resolution: 256 x 256 px (#1 class of background images "Background_without_leaves": 256 x 192 px)

-ix- classes/labels:

PnDC01. Apple_scab

PnDC02. Apple_black_rot

PnDC03. Apple_cedar_apple_rust

PnDC04. Apple_healthy

PnDC05. Background_without_leaves

PnDC06. Blueberry_healthy

PnDC07. Cherry_powdery_mildew

PnDC08. Cherry_healthy

PnDC09. Corn_gray_leaf_spot

PnDC10. Corn_common_rust

PnDC11. Corn_northern_leaf_blight

PnDC12. Corn_healthy

PnDC13. Grape_black_rot

PnDC14. Grape_black_measles

PnDC15. Grape_leaf_blight

PnDC16. Grape_healthy

PnDC17. Orange_haunglongbing

PnDC18. Peach_bacterial_spot

PnDC19. Peach_healthy

PnDC20. Pepper_bacterial_spot

PnDC21. Pepper_healthy

PnDC22. Potato_early_blight

PnDC23. Potato_healthy

PnDC24. Potato_late_blight

PnDC25. Raspberry_healthy

PnDC26. Soybean_healthy

PnDC27. Squash_powdery_mildew

PnDC28. Strawberry_healthy

PnDC29. Strawberry_leaf_scorch

PnDC30. Tomato_bacterial_spot

PnDC31. Tomato_early_blight

PnDC32. Tomato_healthy

PnDC33. Tomato_late_blight

PnDC34. Tomato_leaf_mold

PnDC35. Tomato_septoria_leaf_spot

PnDC36. Tomato_spider_mites_two-spotted_spider_mite

PnDC37. Tomato_target_spot

PnDC38. Tomato_mosaic_virus

PnDC39. Tomato_yellow_leaf_curl_virus

-x- short evaluation of the image contents:

---1- mostly homogeneous background - highly contrasted color with respect to the foreground, sometimes mixed with shaded regions, sometimes no background (e.g. "Corn_healthy" class)

---2- foreground represents plant leaves (true leaves as opposed to embryonic seed leaves)

---3- variance in image resolutions is rather low

---4- no common plant species with the dataset I (i.e. "species")

---5- "Background_without_leaves" class/label is a complete outlier in almost every sense, with no evident characteristic pattern(s) across class images

---6- certain plant species have no "_healthy"-labeled images (e.g. "Orange"), certain have only "_healthy"-labeled images (e.g. "Soybean")

---7- certain plant diseases are present for several plant species (e.g. "_black_rot")

Concluding remark on building classes/labels: the resolution of the matter on "as is" adoption of identification classes and derivation of new ones is still pending

4. What analysis needs to be done?

I. Since the present Data Science Project represents beyond any reasonable doubt a solution to a certain type of image classification, it could be unequivocally attributed to the field of Deep Learning, and, even more specifically, to the use cases of specific Neural Networks that deal with images, the two most widespread representatives being CNN and DNN. Although some studies verify the tendency to opt for CNN as the best choice, the final decision on choosing between building a CNN vs DNN is yet to be made, as well as the pick of a specific Python library for implementation purposes.

II. Although not being a primary mean, employing so called "Transfer Learning" approach of reusing pretrained state-of-the-art models with the positive track record (provided they qualify for solving problems of our kind and the necessary measures are undertaken to preprocess the input accordingly) might be taken into consideration for performance comparison purposes, e.g. as a reference point. Such pretrained models constitute an integral part of some relevant Python libraries like ***Keras***.

III. Evaluation criteria (upon running the model(s) fitted using the train data on the test data):

O. twofold, based on the mean [in compliance with the AI-challenge event regulations]:

-1- F1 score:

---i- target value: the closer it goes up to 1, the better

---ii- formula:

------1- for every class, F1 score represents the harmonic mean of precision and recall, i.e.

F1_score = 2 * (precision * recall)/(precision + recall), whereas

precision = tp/(tp + fp) and

recall = tp/(tp + fn), whereas

tp stands for "true positives", fp - for "false positives" and fn denotes "false negatives"

remark: "positive" or "1" means belonging to the target class, whereas

"negative" or "0" goes for not belonging to the target class

------2- finally, the mean F1 score over all classes is computed

-2- Log Loss:

---i- target value: the closer it comes down to 0, the better

---ii- formula: the negative average over all test set's images i of the sum of y_ij * ln(p_ij) products running over all classes j, whereas

y_ij = boolean value representing if the i-th image in the test set belongs to the j-th class, and

p_ij = probability, predicted/computed by applying our training model to the test set image i, that it belongs to the j-th class

remark: the reason of taking the negative is due to the fact that the (natural) logarithmic function ln() is negative for arguments < 1,

and therefore the negation flips these negative values to positive ones

5. References:

1. The layout of the present document is based on the following blog post of Ekaterina Novoseltseva, hosted by the Tech Hub in Barcelona:
   https://apiumhub.com/tech-blog-barcelona/scoping-data-science-projects/
   Accessed on 11.05.2024

2. The Plant Seedlings Dataset:
   https://vision.eng.au.dk/plant-seedlings-dataset/
   Accessed on 11.05.2024

3. The PlantVillage Dataset:
   https://data.mendeley.com/datasets/tywbtsjrjv/1
   Accessed on 11.05.2024

4. Comparing Image Classification with Dense Neural Network and Convolutional Neural Network – performance case study by Muhammad Adisatriyo Pratama:
   https://medium.com/analytics-vidhya/comparing-image-classification-with-dense-neural-network-and-convolutional-neural-network-5f376582a695
   Accessed on 11.05.2024

5. PlantVillage Disease Classification Challenge:
   https://www.aicrowd.com/challenges/plantvillage-disease-classification-challenge
   Accessed on 11.05.2024