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 ratio ~1.0x)

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

-ix- classes/labels:

PSC01. Maize	PSC07. Cleavers
PSC02. Common wheat	PSC08. Charlock
PSC03. Sugar beet	PSC09. Fat Hen
PSC04. Scentless Mayweed	PSC10. Small-flowered Cranesbill
PSC05. Common Chickweed	PSC11. Black-grass
PSC06. Shepherd's Purse	PSC12. Loose Silky-bent

-x- short evaluation of the image contents:

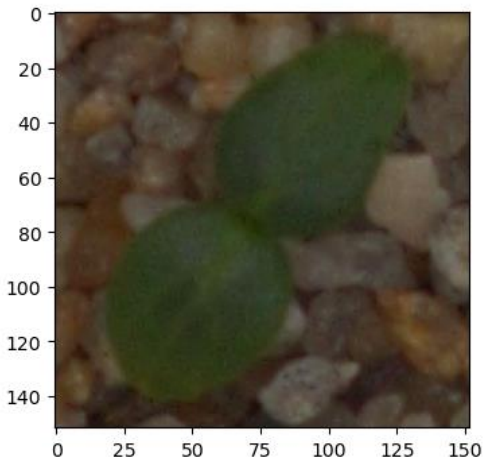
---1- mostly heterogeneous background - brownish granular texture

---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")

-xi- representative image of the dataset [seedling of the plant species "Cleavers"]:



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: JPEG 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	PnDC20. Pepper_bacterial_spot
PnDC02. Apple_black_rot	PnDC21. Pepper_healthy
PnDC03. Apple_cedar_apple_rust	PnDC22. Potato_early_blight
PnDC04. Apple_healthy	PnDC23. Potato_healthy
PnDC05. Background_without_leaves	PnDC24. Potato_late_blight
PnDC06. Blueberry_healthy	PnDC25. Raspberry_healthy
PnDC07. Cherry_powdery_mildew	PnDC26. Soybean_healthy
PnDC08. Cherry_healthy	PnDC27. Squash_powdery_mildew
PnDC09. Corn_gray_leaf_spot	PnDC28. Strawberry_healthy
PnDC10. Corn_common_rust	PnDC29. Strawberry_leaf_scorch
PnDC11. Corn_northern_leaf_blight	PnDC30. Tomato_bacterial_spot
PnDC12. Corn_healthy	PnDC31. Tomato_early_blight
PnDC13. Grape_black_rot	PnDC32. Tomato_healthy
PnDC14. Grape_black_measles	PnDC33. Tomato_late_blight
PnDC15. Grape_leaf_blight	PnDC34. Tomato_leaf_mold
PnDC16. Grape_healthy	PnDC35. Tomato_septoria_leaf_spot
PnDC17. Orange_haunglongbing	PnDC36. Tomato_spider_mites_two-spotted_spider_mite
PnDC18. Peach_bacterial_spot	PnDC37. Tomato_target_spot
PnDC19. Peach_healthy	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

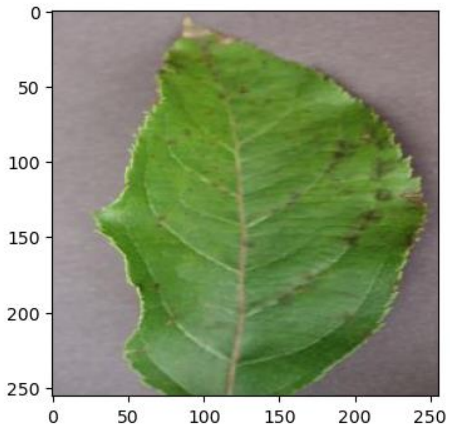
---4- no common plant species with the dataset I (i.e. "species")

---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")

-xi- representative image of the dataset [“scab”-diseased mature leaf of the plant species “Apple”]:



-xii- series of grayscale components of “RGB” channels of the representative image:

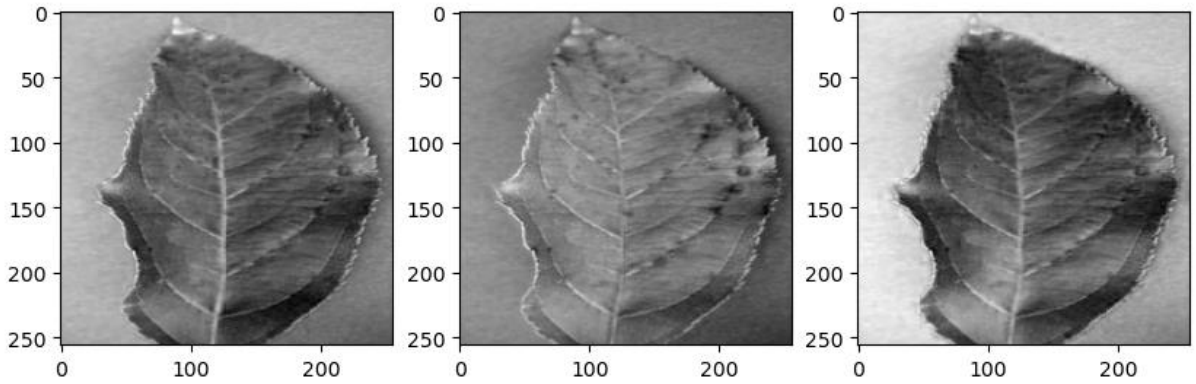


Figure 1: the middle image corresponds to the "Green" channel, expectedly exhibiting larger i.e. lighter pixel intensities over the leaf surface

-xiii- 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. Data Audit

	Number of lines in the table : *60987 (excluding header)*							
# Column	**Name of the Column**	**Variable's type**	**Description**	**Is the variable available before prediction**	**Variable's type**	**Percentage of missing values**	**Categorical / Quantitative**	**Distribution**
		Is the variable a feature or the target ? (Only applicable for supervised learning projects)	*What does this variable represent (from a business perspective ?)*	*Is this variable known before the prediction is made ? (Only applicable for supervised learning projects)*	*int64, float etc...* *If "object", develop.*	*in %*		*For categorical variables with less than 10 categories, list all categories.* *For quantitative variables, detail the distribution (basic descriptive statistics)*
1	folder_name	N/A (DL OR Project)	contextually: label [species/background] [___healthy/disease]	N/A (DL OR Project)	object:str	0,00%	Categorical - more than 10 categories	see GitHub notebooks
2	file_name	N/A (DL OR Project)	image file name	N/A (DL OR Project)	object:str	0,00%	Unique Value	see GitHub notebooks
3	width_px	N/A (DL OR Project)	image width in px	N/A (DL OR Project)	int64	0,00%	Quantitative	see GitHub notebooks
4	height_px	N/A (DL OR Project)	image height in px	N/A (DL OR Project)	int64	0,00%	Quantitative	see GitHub notebooks
5	bits_p_px	N/A (DL OR Project)	color depth per px (in bits)	N/A (DL OR Project)	int64	0,00%	Quantitative	see GitHub notebooks
6	px_format	N/A (DL OR Project)	image color model	N/A (DL OR Project)	object:str	0,00%	Categorical - 3 to 5 categories	see GitHub notebooks
7	mime	N/A (DL OR Project)	image MIME-format	N/A (DL OR Project)	object:str	0,00%	Categorical - Binary	see GitHub notebooks
8	channels	N/A (DL OR Project)	number of image channels	N/A (DL OR Project)	int64	0,00%	Quantitative	see GitHub notebooks
9	chn_0_px_std	N/A (DL OR Project)	standard deviation of image px intensities (1st channel)	N/A (DL OR Project)	float64	0,00%	Quantitative	see GitHub notebooks
10	chn_0_px_min	N/A (DL OR Project)	minimum of image px intensities (1st channel)	N/A (DL OR Project)	float64	0,00%	Quantitative	see GitHub notebooks
11	chn_0_px_q1	N/A (DL OR Project)	1st quartile of image px intensities (1st channel)	N/A (DL OR Project)	float64	0,00%	Quantitative	see GitHub notebooks
12	chn_0_px_med	N/A (DL OR Project)	median of image px intensities (1st channel)	N/A (DL OR Project)	float64	0,00%	Quantitative	see GitHub notebooks
13	chn_0_px_avg	N/A (DL OR Project)	mean of image px intensities (1st channel)	N/A (DL OR Project)	float64	0,00%	Quantitative	see GitHub notebooks
14	chn_0_px_q3	N/A (DL OR Project)	3rd quartile of image px intensities (1st channel)	N/A (DL OR Project)	float64	0,00%	Quantitative	see GitHub notebooks
15	chn_0_px_max	N/A (DL OR Project)	maximum of image px intensities (1st channel)	N/A (DL OR Project)	float64	0,00%	Quantitative	see GitHub notebooks
16	chn_0_px_sum	N/A (DL OR Project)	sum of image px intensities (1st channel)	N/A (DL OR Project)	int64	0,00%	Quantitative	see GitHub notebooks
17	chn_1_px_std	N/A (DL OR Project)	standard deviation of image px intensities (2nd channel)	N/A (DL OR Project)	float64	0,00%	Quantitative	see GitHub notebooks
18	chn_1_px_min	N/A (DL OR Project)	minimum of image px intensities (2nd channel)	N/A (DL OR Project)	float64	0,00%	Quantitative	see GitHub notebooks
19	chn_1_px_q1	N/A (DL OR Project)	1st quartile of image px intensities (2nd channel)	N/A (DL OR Project)	float64	0,00%	Quantitative	see GitHub notebooks
20	chn_1_px_med	N/A (DL OR Project)	median of image px intensities (2nd channel)	N/A (DL OR Project)	float64	0,00%	Quantitative	see GitHub notebooks
21	chn_1_px_avg	N/A (DL OR Project)	mean of image px intensities (2nd channel)	N/A (DL OR Project)	float64	0,00%	Quantitative	see GitHub notebooks
22	chn_1_px_q3	N/A (DL OR Project)	3rd quartile of image px intensities (2nd channel)	N/A (DL OR Project)	float64	0,00%	Quantitative	see GitHub notebooks
23	chn_1_px_max	N/A (DL OR Project)	maximum of image px intensities (2nd channel)	N/A (DL OR Project)	float64	0,00%	Quantitative	see GitHub notebooks
24	chn_1_px_sum	N/A (DL OR Project)	sum of image px intensities (2nd channel)	N/A (DL OR Project)	int64	0,00%	Quantitative	see GitHub notebooks
25	chn_2_px_std	N/A (DL OR Project)	standard deviation of image px intensities (3rd channel)	N/A (DL OR Project)	float64	0,00%	Quantitative	see GitHub notebooks
26	chn_2_px_min	N/A (DL OR Project)	minimum of image px intensities (3rd channel)	N/A (DL OR Project)	float64	0,00%	Quantitative	see GitHub notebooks
27	chn_2_px_q1	N/A (DL OR Project)	1st quartile of image px intensities (3rd channel)	N/A (DL OR Project)	float64	0,00%	Quantitative	see GitHub notebooks
28	chn_2_px_med	N/A (DL OR Project)	median of image px intensities (3rd channel)	N/A (DL OR Project)	float64	0,00%	Quantitative	see GitHub notebooks
29	chn_2_px_avg	N/A (DL OR Project)	mean of image px intensities (3rd channel)	N/A (DL OR Project)	float64	0,00%	Quantitative	see GitHub notebooks
30	chn_2_px_q3	N/A (DL OR Project)	3rd quartile of image px intensities (3rd channel)	N/A (DL OR Project)	float64	0,00%	Quantitative	see GitHub notebooks
31	chn_2_px_max	N/A (DL OR Project)	maximum of image px intensities (3rd channel)	N/A (DL OR Project)	float64	0,00%	Quantitative	see GitHub notebooks
32	chn_2_px_sum	N/A (DL OR Project)	sum of image px intensities (3rd channel)	N/A (DL OR Project)	int64	0,00%	Quantitative	see GitHub notebooks

Figure 2: descriptive analysis of the working .csv data file containing metadata as well as the most common per-channel pixel-based statistics, extracted from the raw images (both datasets)

5. Data Exploration

Exploration of technical attributes

Our consolidated dataset contains ~61K pictures. We already saw that they came from different sources. So, we must assume that they are not standardized. We need to understand whether this results into specific modelling requirements. The exploration therefore starts with an analysis of their technical attributes followed by an analysis on the semantic content.

Image formats

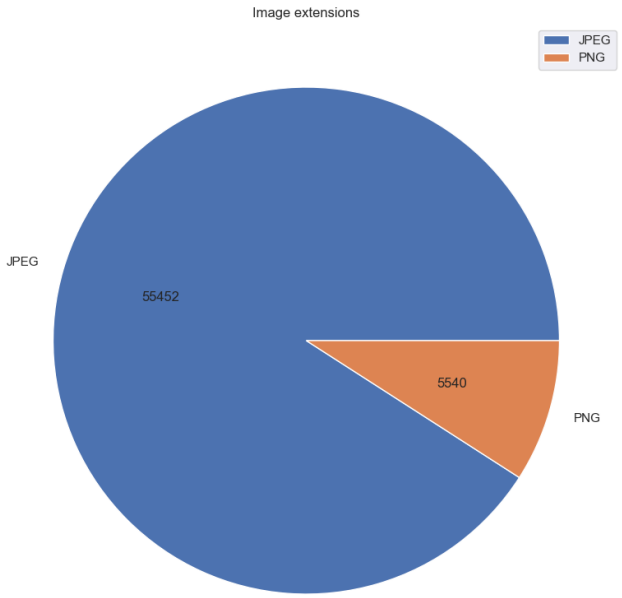


Figure 3: pie chart showing the distribution of image formats in absolute frequencies

As already seen, the data comes with different image extensions. The majority of the data is in JPEG format while a smaller part is in the format PNG. The two formats are quite similar as both can support up to 16 million colors. Also, both formats use compression. The main difference however is that the PNG format is using lossless compression while the JPEG format is using lossy compression. Still the JPEG format is commonly used especially for photography.

We will check during the modelling phase whether there is a relevance from having these different formats.

Image color models

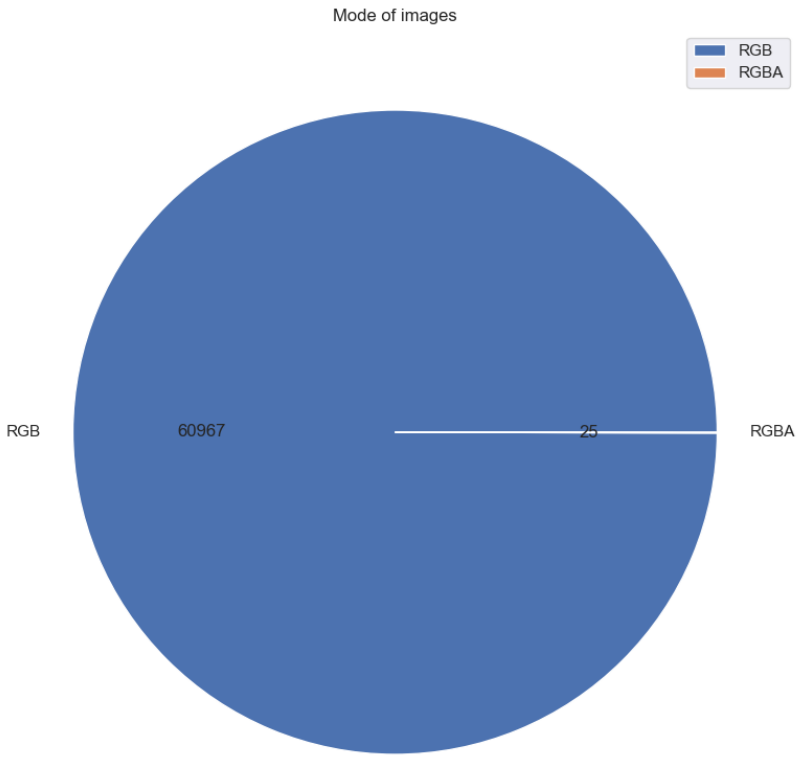


Figure 4: pie chart showing the distribution of image color models in absolute frequencies

Almost all pictures have the color model "RGB". However, there is a handful exceptions with pictures of the color model "RGBA", which represents just an extension of “RGB” by adding the opaqueness attribute to pixels.

The difference is that "RGB" is a 3-channel format while "RGBA" has 4 channels. This means that we will need to support both modes. However, as the number of "RGBA" pictures is very low, there would also the option to discard them from the dataset.

We will check this in the modelling phase of the project.

Shape of images (width & height)

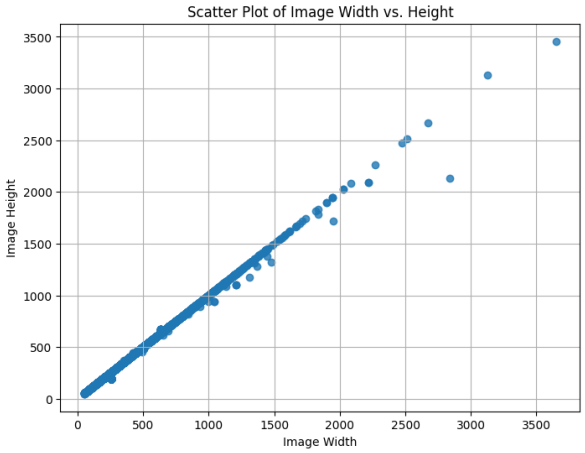


Figure 5: scatter plot of width vs height in pixels

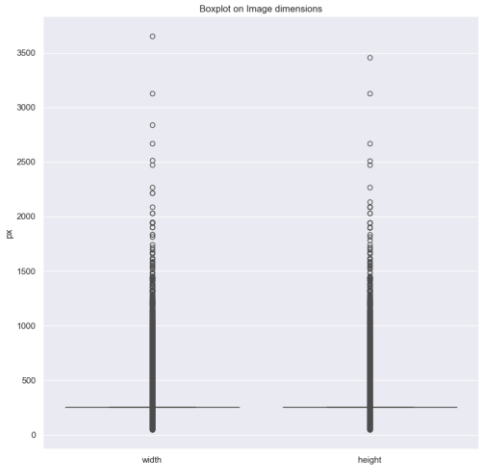


Figure 6: aligned dot plot of image width & height (in pixels) with median-whiskers

As we can see there is a wide range of shapes on the pictures. The analysis shows that most of the pictures are square, but there are exceptions.

Also, we see from the boxplot that the vast majority of the pictures has a size of ca. 250 x 250. However, there are several extreme values outside of the IQR (inter quartile range) of this data range.

We may need to assume that we will need to use harmonized shapes in the data model. That would mean that we would need to apply a related harmonization step for the data.

Exploration of picture semantics

In parallel to the technical attributes, we also analyze the semantic content of the pictures.

Plant species

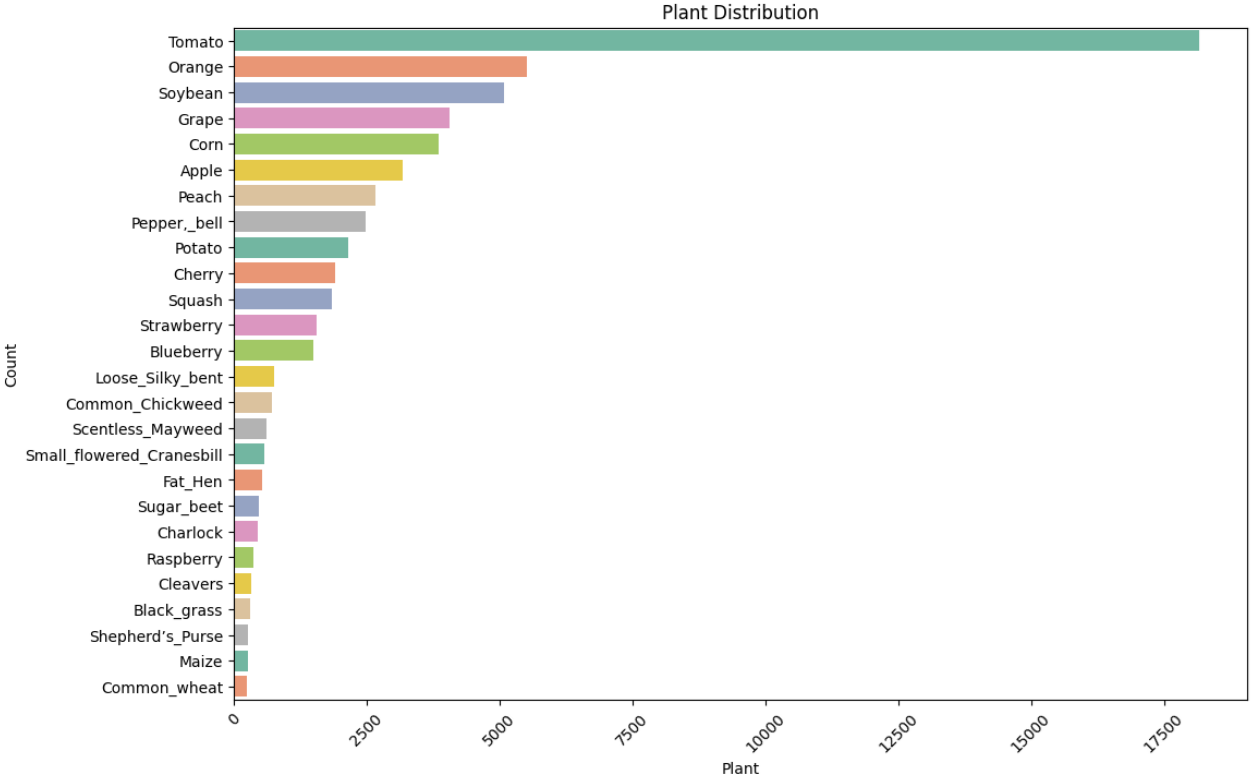


Figure 7: horizontally oriented bar chart showing the distribution of images of different plant species in absolute frequencies

We start with analyzing the different plants that are available in the dataset. In total we have 26 different plants. The most prominent plant is the tomato which represents almost 30% of the dataset.

We will check during the modelling phase whether this uneven distribution generates an impact.

Diseases of plants

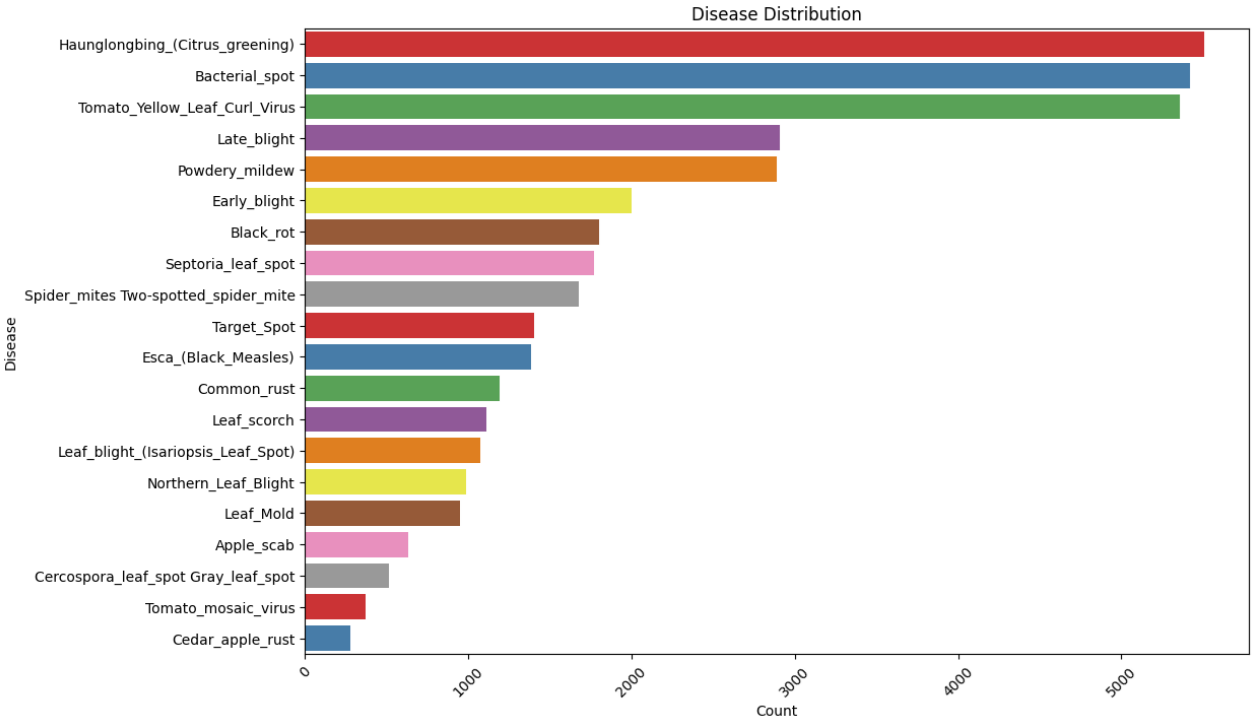


Figure 8: horizontally oriented bar chart showing the distribution of images with different plant diseases in absolute frequencies

In total our dataset shows 20 different types of diseases. The distribution by disease is also quite different and this also may have an impact on the model.

Healthy vs diseased plants

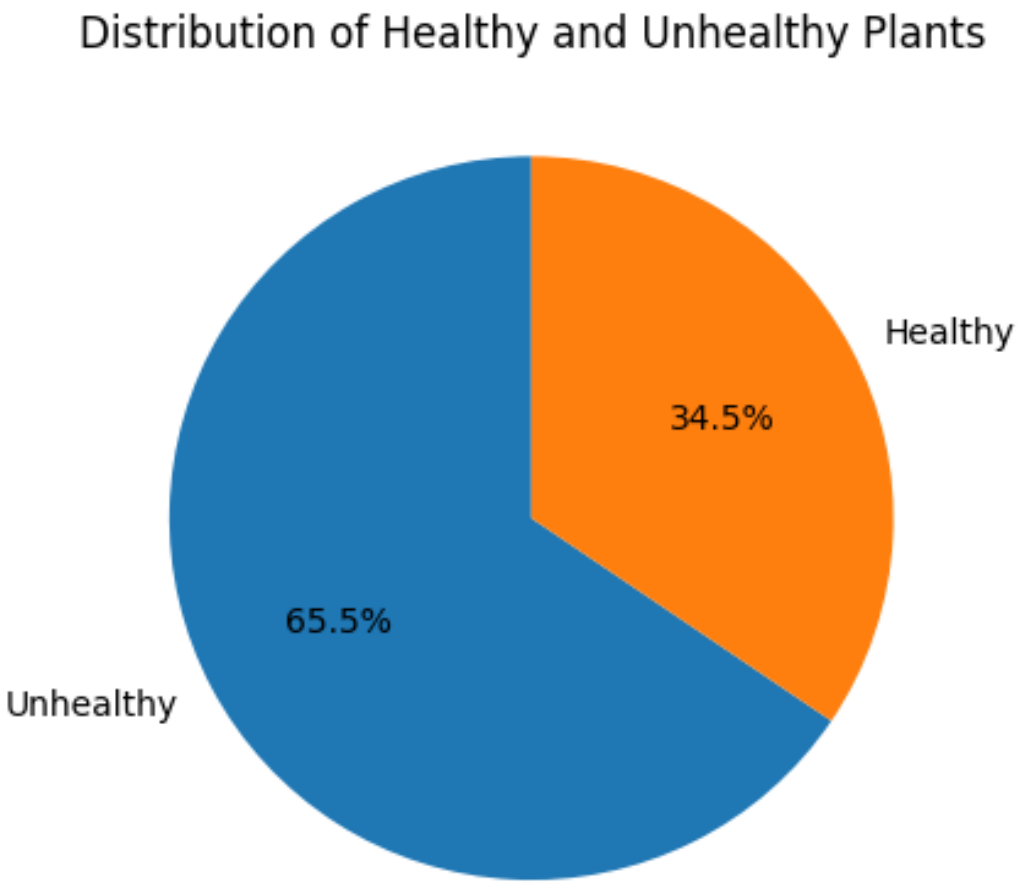


Figure 9: pie chart showing the distribution of images with healthy vs diseased plants in relative frequencies

Our dataset contains pictures of healthy and of unhealthy plants. Around 2/3 of the pictures show unhealthy plants. This means that there is ca. 1/3 of the dataset related to plants classified as healthy.

Pairwise relationships of pixel-based statistics

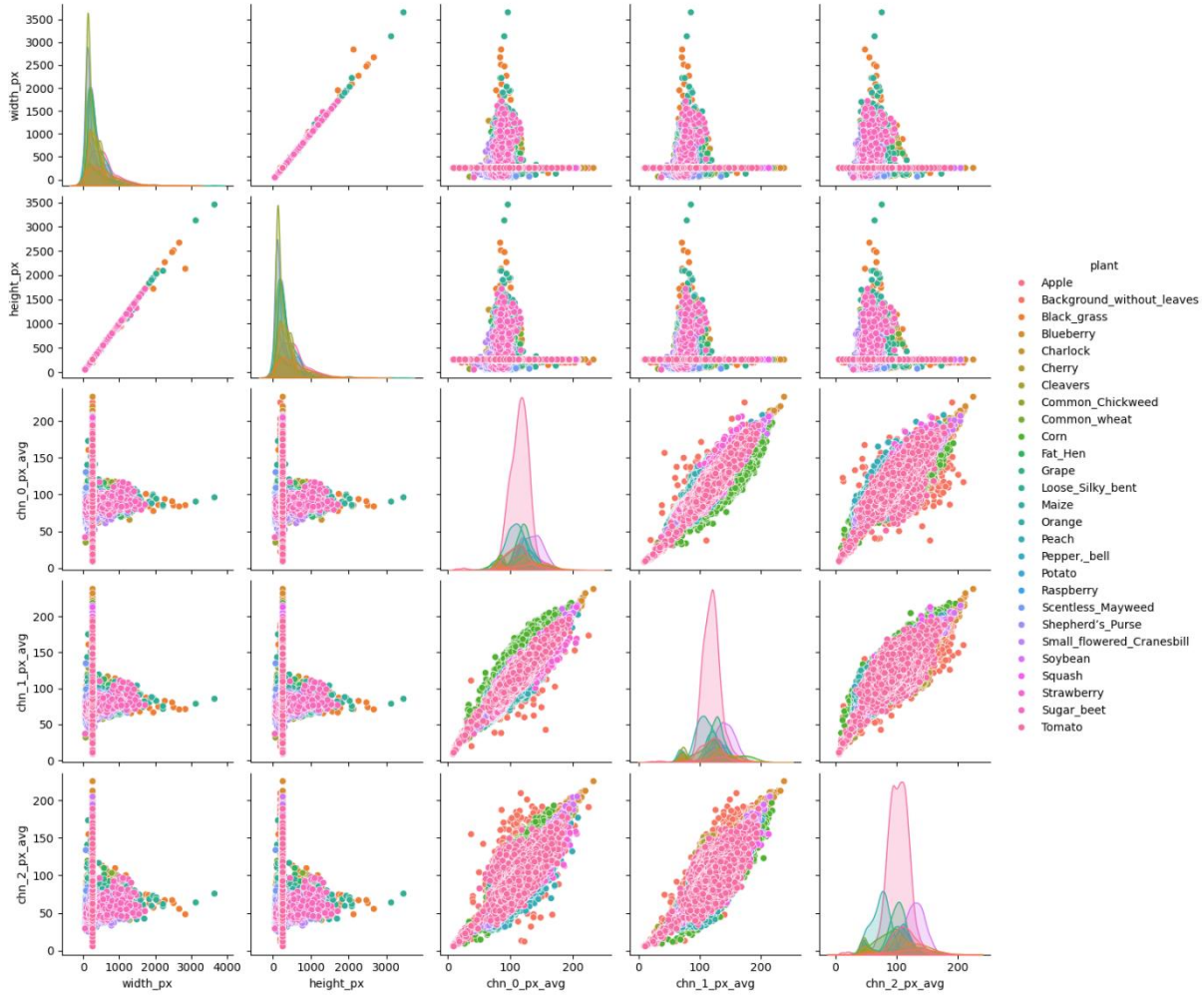


Figure 10: pairplot showing the grid of scatterplots and KDE diagonal plots of image width, height, and per-channel means in pixels

The diagonal plots represent kernel density estimate curves of 5 chosen (not cherry-picked) pixel-based features (two metadata elements and three pixel-based statistics – in this case, raw empirical statistical moments of first order i.e. sample means of pixel intensities per each of "RGB" channels). Whereas the skewed distribution of image width and height does not surprise us (due to the analysis conducted in in previous visualizations), the apparent indication of bimodality for the mean pixel intensity distributions of predominantly "Blue" channel ("Green" is also affected, but not as much) might need to be addressed in our future endeavors.

Scatterplots of mean pixel intensities between different channels demonstrate pretty strong positive linear correlation among plant species, as opposed to the class "Background_without_leaves" of images that in fact represent a pack of various indoor and outdoor scenery images. This class (the dots of orange hue) exhibits the significant absence of linear correlation between pixel intensity means of any pair of "RGB" channels what is perfectly explainable via the intrinsic large-scale heterogeneity of versatile scenery images.

The analysis on ways of treating this outlier class of images is still pending.

6. 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

7. 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