Hunger for none

Bayer 2021 Hackathon
solution by Adrian Brodzik

Objectives

Binary classifier

Determine whether a plant is healthy or diseased

Multi-class classifier

Identify the specific plant disease

User interface

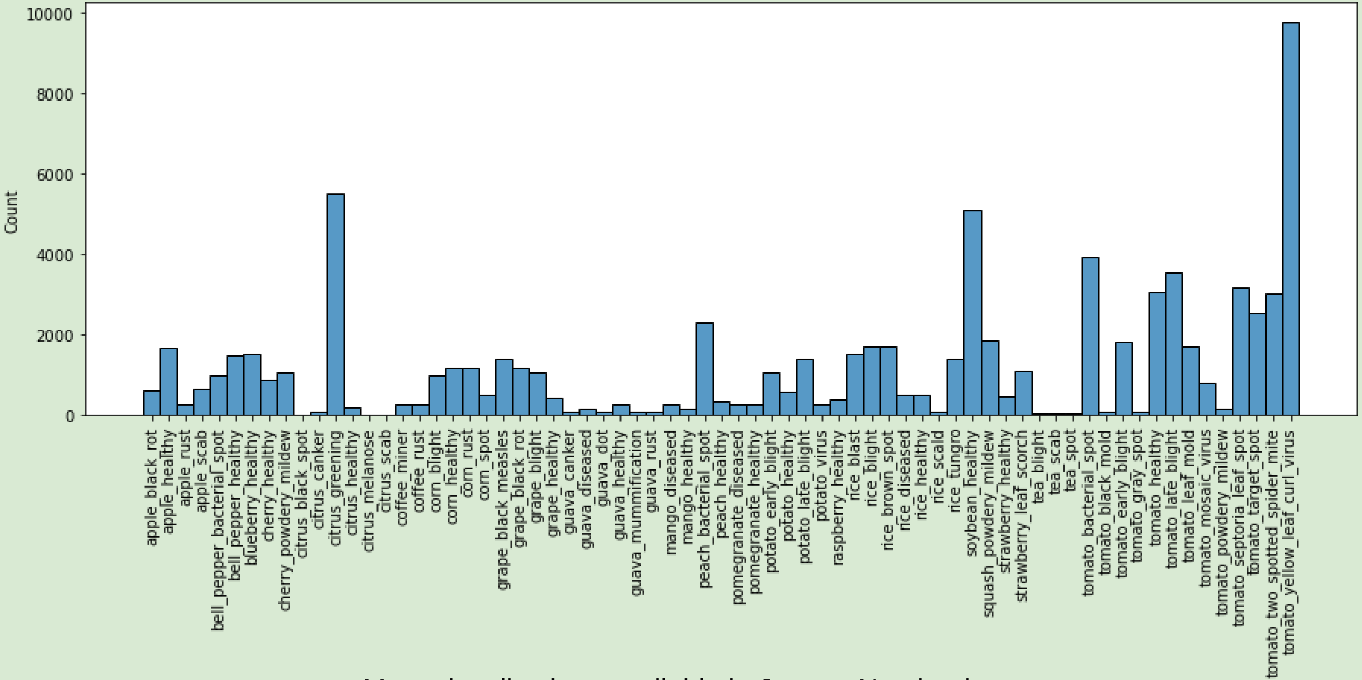
Serve predictions to the end user

Data preparation

- The foundation of every data science project is a **good dataset**.
- By combining different datasets from different sources we can achieve adequate data diversity and reduce the risk of overfitting.
- The final dataset has over **80k images** of plants, namely their leaves. All relevant sources have been listed in the bibliography.
- This is a good start but more data would be beneficial, especially images of fruits and vegetables. In a real-world scenario there would be a **feedback loop**. Users, by running our app, would be expanding our dataset with their images of crops, improving the overall accuracy of the predictors for everyone.
- The distribution and **class imbalance** of our data may be problematic.
- Disease class labels are **sparse** (mostly zero), which may be problematic.

Data preparation

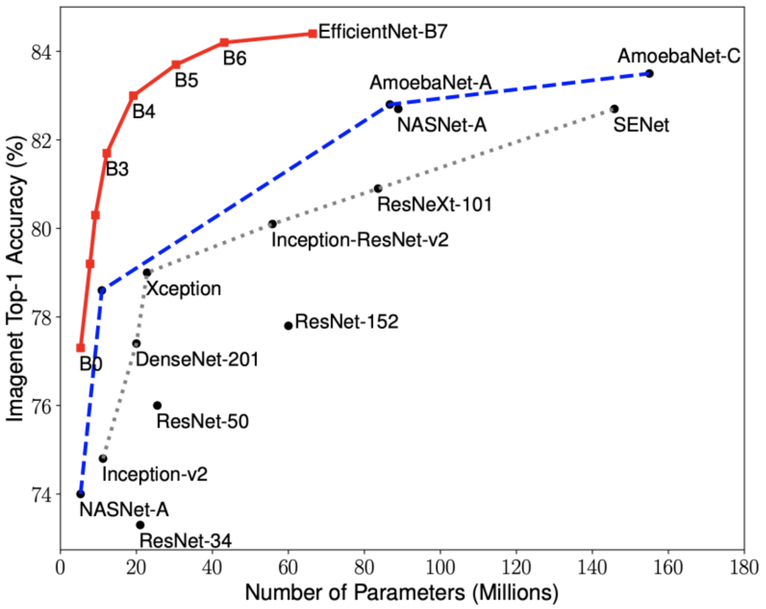
(plant, condition) distribution



More visualizations available in Jupyter Notebooks.

Model training

- The solution is based on the **EfficientNet** convolutional neural network architecture.
- Model architecture is a personal preference.
- Even the most sophisticated model would be useless without an appropriate dataset.
- The loss function is **binary cross-entropy**.
- The optimizer is **Adam** with weight decay.
- Train/dev/test data split is 60%-20%-20%.
- The evaluation metric is **F_1 score**.
- Images are resized to **256x256 pixels**.



model size vs. ImageNet accuracy
(source: https://arxiv.org/abs/1905.11946)

Better generalization

data sampler

weigh samples inversely to class appearing probability

augmentations

random image flips, rotations, brightness, contrast, saturation, blur

CutMix

swap portions of images and their labels proportionally

early stopping

stop training if loss starts to increase

weighted class loss

trade off recall and precision by adding weights to positive samples

variable learning rate

decrease the learning rate with each epoch

Results & Conclusions

- Both models perform well on the dataset but may run into problems when presented out-of-distribution images, e.g. Google Images.
- The binary classification model has a suspiciously high F_1 score. This may be the result of a **data leak** (the model learned some feature that it wasn't supposed to, e.g. background, hidden watermark).
- Further training and dataset expansion is required to make this a commercially sustainable product.
- Simple single model solutions based on: EfficientNet B4 and B5.
- Trained using limited resources: Kaggle and Google Colab notebooks.

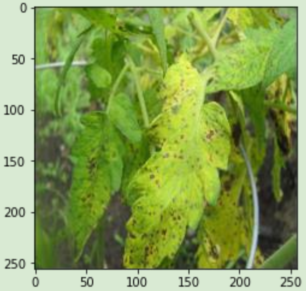
binary classification model

	BCE loss	F_1 micro	F_1 macro
train	0.0013	0.9997	0.9996
dev	0.0079	0.9977	0.9969
test	0.0086	0.9976	0.9966

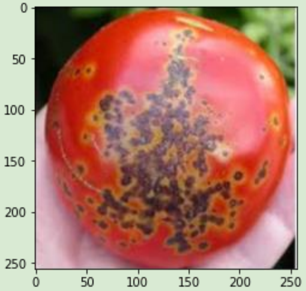
disease identification model

	BCE loss	F_1 micro	F_1 macro
train	0.0100	0.9513	0.9411
dev	0.0120	0.9422	0.8362
test	0.0111	0.9464	0.8653

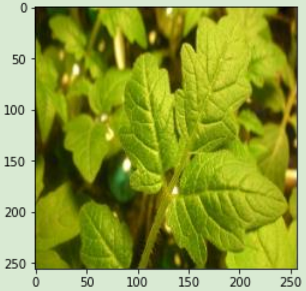
Sample binary predictions



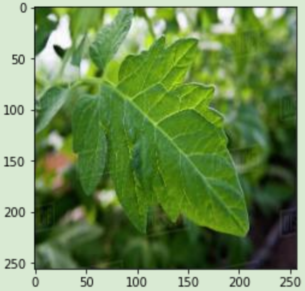
0% healthy



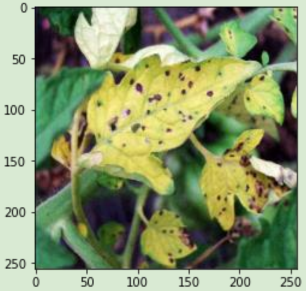
0% healthy



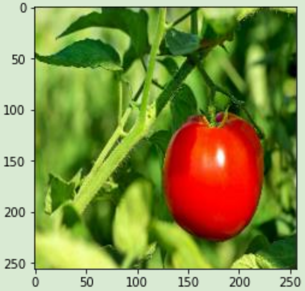
97% healthy



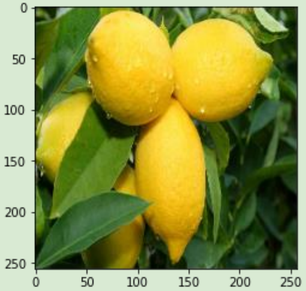
99% healthy



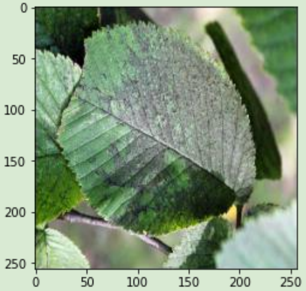
13% healthy



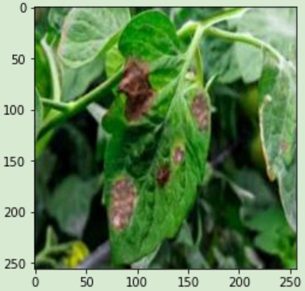
99% healthy



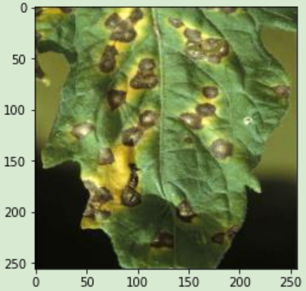
99% healthy



99% healthy



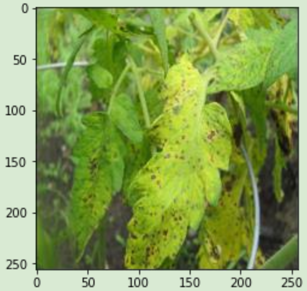
0% healthy



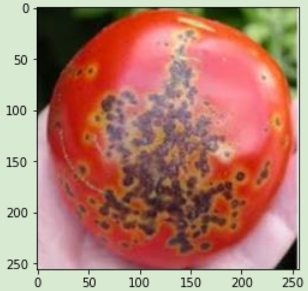
46% healthy

source: Google Images

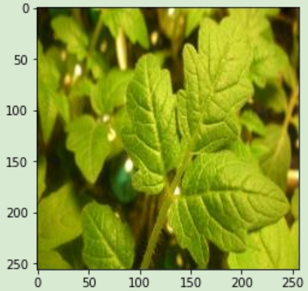
Sample disease identification predictions



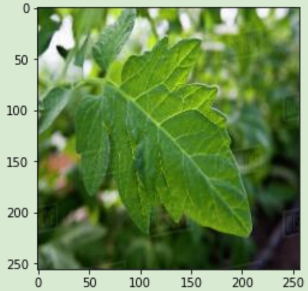
74% melanose



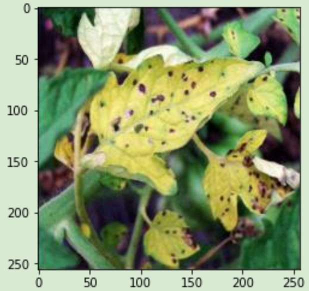
78% leaf scorch



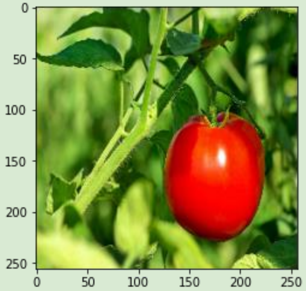
70% healthy



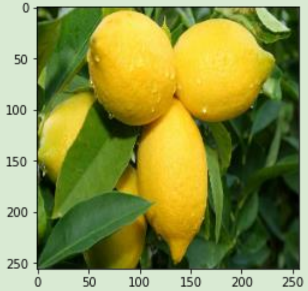
41% healthy



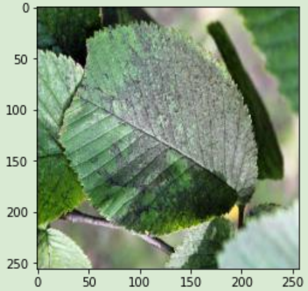
52% healthy



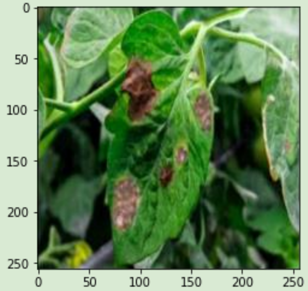
38% healthy



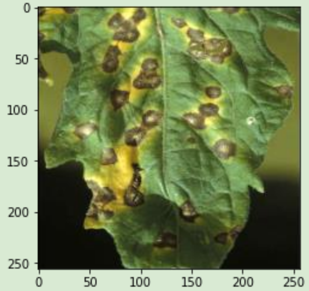
63% healthy



21% healthy



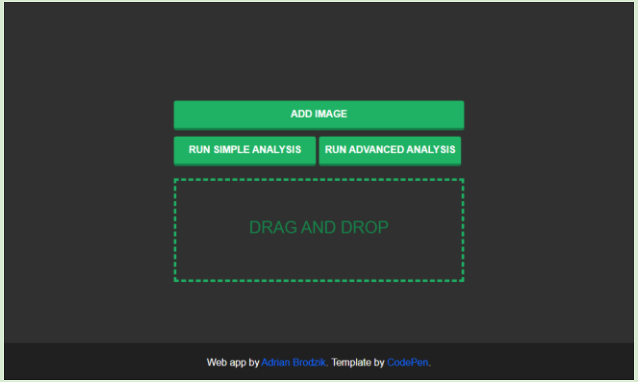
28% gray spot



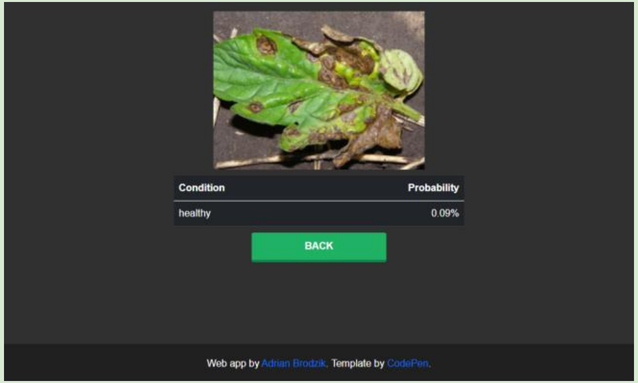
7% late blight

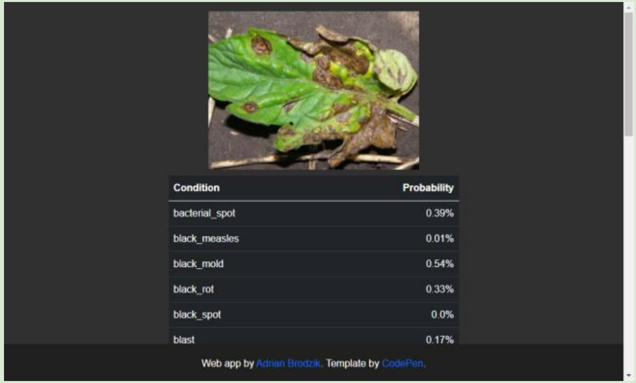
source: Google Images

User interface



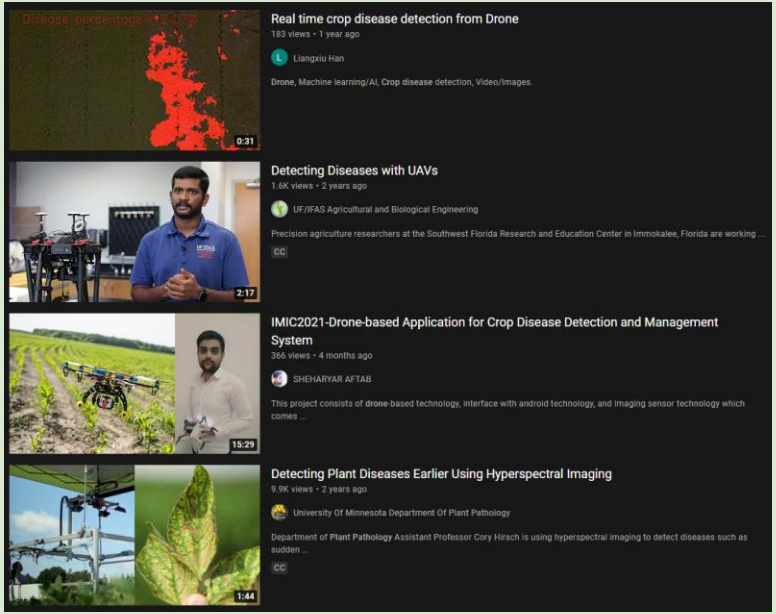
- This is a proof-of-concept. There is much room for improvement.
- Simple, lightweight web application.
- Powered by Flask and Bootstrap.
- Example usage in the video.





Future development

- Improving out-of-distribution predictions by diversifying the dataset.
- Creating disease identification models for individual crops, since different plants can have different non-overlapping diseases.
- Improving user interface. Developing a fully-fledged web application. Let users analyze multiple images at once.
- IoT based solution where drones, tractors, smartphones can send images to a local server for processing.
- Using drones and binary classification models to identify region borders of diseased plants. If one plant is infected, then there must be more.



existing disease identification solutions (source: YouTube)

Future development

- Switch objective from classification to **anomaly detection**. Farmers and gardeners are very experienced in the life cycle of their plants and potential pathogens, therefore it may be unnecessary to create an advanced model to identify every single disease.
- Instead search the plants for anomalies and notify the farmer with an image via their smartphone, e.g. from a drone scouting the field.
- Advanced disease identification is not necessary but would be a “quality of life” tool. Alternatively, a so called **expert system** could suggest ways to combat an identified pathogen.

Thank you.

Bibliography

- Presentation theme: https://slidesgo.com/theme/inspirational-green
- https://www.kaggle.com/vipoooool/new-plant-diseases-dataset
- https://data.mendeley.com/datasets/ngdgg79rzb
- https://data.mendeley.com/datasets/v4w72bsts5
- https://data.mendeley.com/datasets/s8x6jn5cvr
- https://www.kaggle.com/c/plant-pathology-2021-fgvc8
- https://github.com/spMohanty/PlantVillage-Dataset/tree/master/raw/color
- https://data.cipotato.org/dataset.xhtml?persistentId=doi:10.21223/IDUWZE
- https://data.cipotato.org/dataset.xhtml?persistentId=doi:10.21223/BCVIZY
- https://data.mendeley.com/datasets/369cky7n39
- https://figshare.com/articles/dataset/Healthy_and_Disease_affected_Leaves_of_Grape_Plant/13083890
- https://data.mendeley.com/datasets/3f83gxmv57
- https://data.mendeley.com/datasets/hb74ynkjcn
- https://data.mendeley.com/datasets/vfxf4trtcg
- https://data.mendeley.com/datasets/dbjyfkn6jr
- https://data.mendeley.com/datasets/znsxdctwtt
- https://data.mendeley.com/datasets/fwcj7stb8r
- https://archive.ics.uci.edu/ml/datasets/Rice+Leaf+Diseases
- https://www.kaggle.com/rajkumar898/rice-plant-dataset
- https://www.kaggle.com/c/plant-pathology-2020-fgvc7