



Plant Health Classification

Ashley Poon
May 13, 2021

Background

In recent years, we've seen huge growth in demand for houseplants, particularly among millennials and Gen Z

- 18 to 34-year-olds now occupy 29% of all gardening households
- Tons of forums dedicated to houseplant care and photography
- Potentially a way to fill the need for nurture
- Provides physical and mental benefits (self-care!)
- Trend fueled by social media





Problem On Hand

Can we classify whether a plant is healthy or unhealthy using object detection?

The goal is to help novice plant owners determine if their plants are healthy or need some special attention.

If the model is able to accurately classify an unhealthy plant, then users can take the next step to diagnose and determine whether their plants are being over/under watered, receiving too much/too little sunlight, infested, etc.



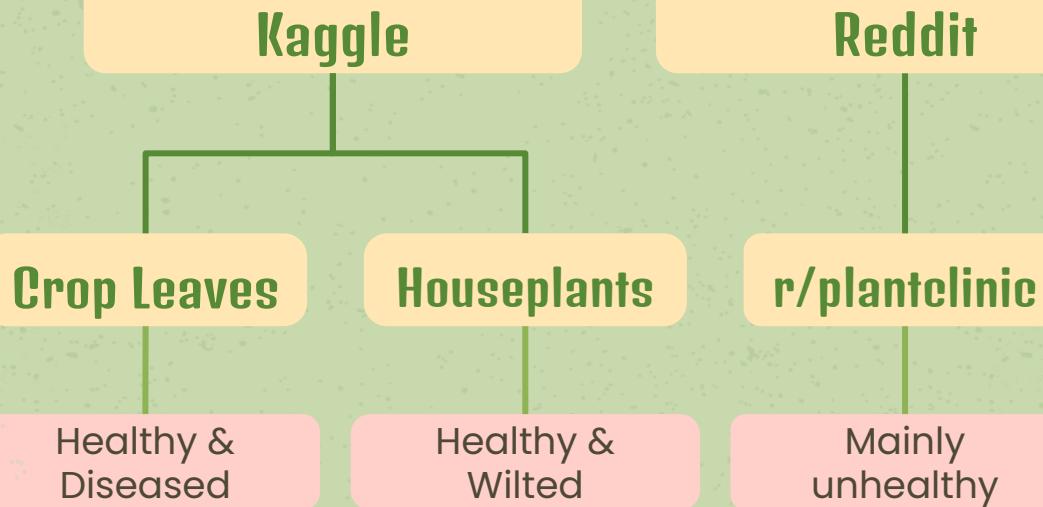
AGENDA

- 01 Data Collection & Cleaning**
- 02 CNN Models - Classify**
- 03 YOLO v4 Model - Detect**
- 04 Demo**
- 05 Conclusion & Next Steps**

01

Data Collection & Cleaning

Data Sources



Crop Leaves

This kaggle dataset contains 88K lab images of healthy and diseased crop leaves. Crops included apple, corn, strawberry, tomato, etc.

Removed poor quality and duplicate images in sample used for modeling.



Houseplants

This kaggle dataset contains 904 Google images of healthy and wilted houseplants.

Minimal cleaning needed as images were already classified and were mostly good quality.



r/plantclinic

Approx. 500 images scraped from the subreddit. The majority of these images submitted by Reddit users are of unhealthy plants.

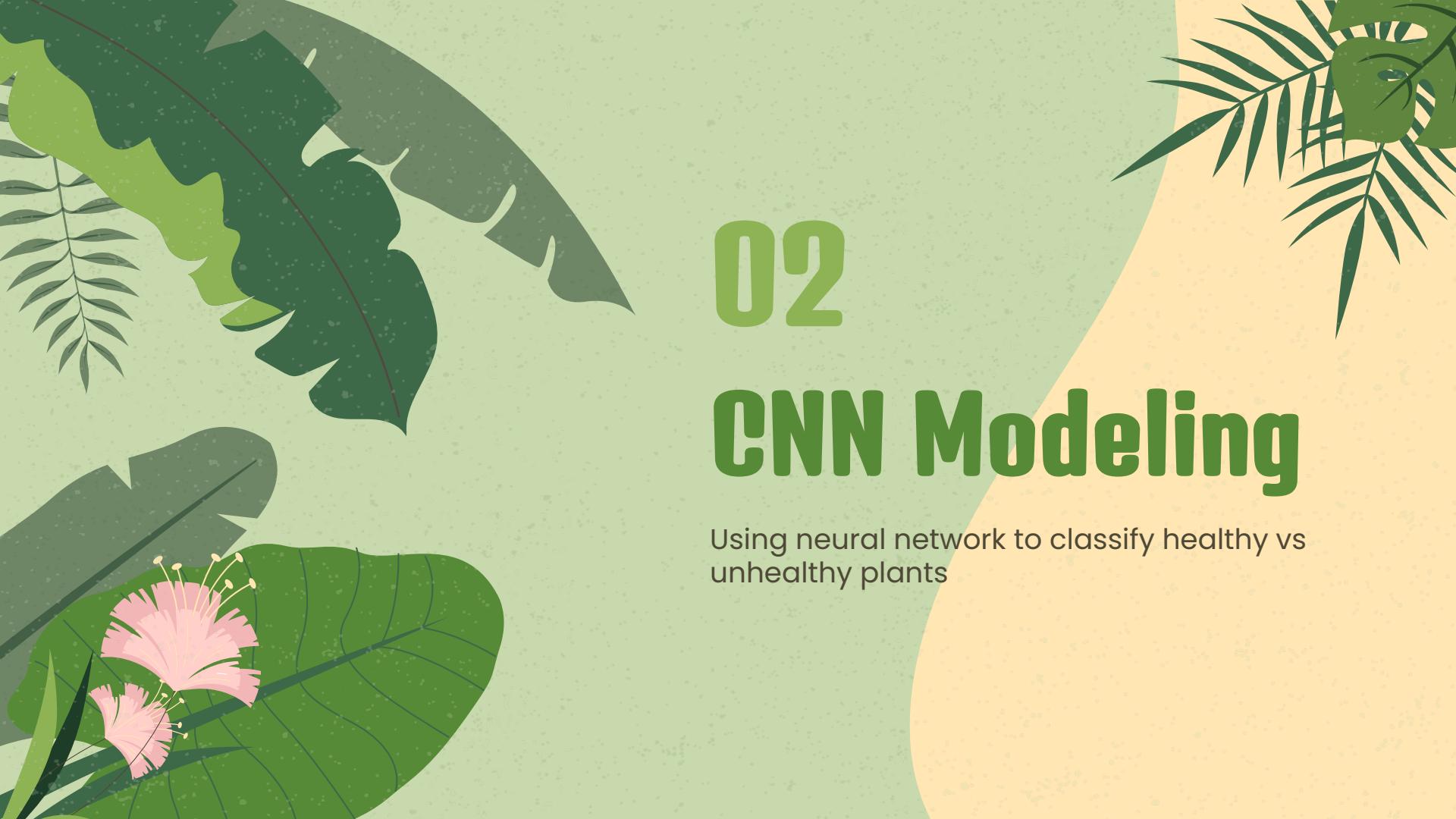
Reviewed each individual picture to confirm plants were indeed unhealthy, removed any low quality or ambiguous pictures.





45,000+

Images available to sample from for modeling.

The background features a lush, tropical-themed collage. On the left, large green leaves with prominent veins are shown, some with small pink flowers at the bottom. On the right, a palm frond with long, thin green leaves is visible against a yellow gradient background.

02

CNN Modeling

Using neural network to classify healthy vs unhealthy plants

Preprocessing

- For the **healthy plant** dataset,
 - Sampled 500 healthy crop leaf images
 - 451 healthy houseplant images
- For the **unhealthy plant** dataset,
 - Sampled 500 unhealthy crop leaf images
 - 451 unhealthy houseplant images
 - 314 Reddit images (for CNN models including Reddit data)
- Processed images for CNN using,
 - **load_img** to resize each image
 - **img_to_array** to convert each image to an array



Initial Model (without Reddit)

Two convolutional layers with 64 filters and one dense layer (excl. output) with 64 nodes.

Layer	Output Shape	Param #
conv2d (Conv2D)	(None, 254, 254, 64)	1,792
max_pooling2d (MaxPooling2D)	(None, 127, 127, 64)	0
conv2d_1 (Conv2D)	(None, 125, 125, 64)	36,928
max_pooling2d_1 (MaxPooling2D)	(None, 62, 62, 64)	0
flatten (Flatten)	(None, 246016)	0
dense (Dense)	(None, 64)	15,745,088
dense_1 (Dense)	(None, 1)	65

Best accuracy score
with minimal overfitting:

78% train | 73% test
(vs 50% baseline)

Iterative Models (with Reddit)



Model 1

Conv layers
(filters):

4 (16 filters)

Dense layers
(nodes):

1 (16 nodes)

Best train accuracy:

79%

Best test
accuracy:

79%

Model 2

4 (16 filters)

1 (64 nodes)

Best accuracy score
with minimal overfitting

Model 3

4 (16 filters)

1 (128 nodes)

85%

80%

Model 4

3 (32 filters)

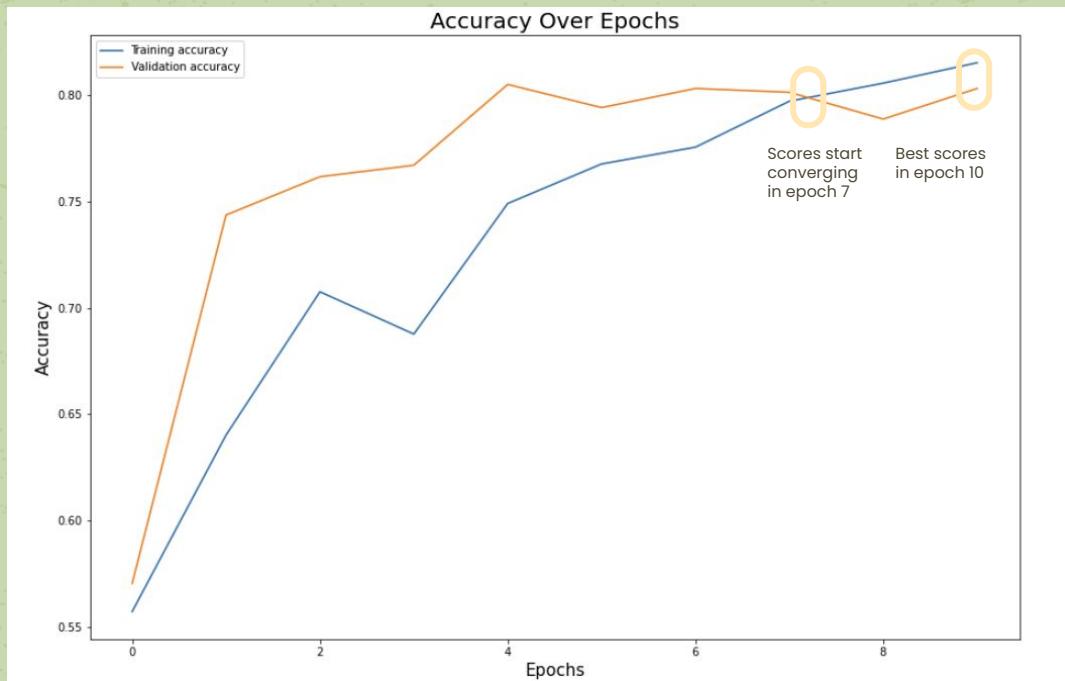
1 (32 nodes)

84%

80%



CNN Model 2 Performance



03

YOLOv4 Modeling

Leveraging computer vision to detect
healthy vs unhealthy plants



YOLO at a High Level

You only look once, a real-time object detection algorithm

1

Dataset

Requires images and yolo-specific labeling and annotations (bounding boxes)

2

Configurations

A cfg file that contains model and training parameters (ie. batch size, filters)

3

Inputs

Includes a names file that contains the class names and a data file that points to the data and backup folder

4

Training

Uses darknet (an open source neural network framework) + the data and cfg files to train

Preprocessing

- For the **healthy plant** dataset, manually labeled and annotated 905 healthy plant images from the crop leaves and houseplant datasets.
- For the **unhealthy plant** dataset, manually labeled and annotated 1,028 unhealthy plant images from the crop leaves, houseplant, and Reddit datasets.
- Defined helper functions to handle images in Colab: display, upload, and download.



Classes & Training Data Composition



Healthy

Crop leaves (457)

Houseplants (448)

TOTAL: 905



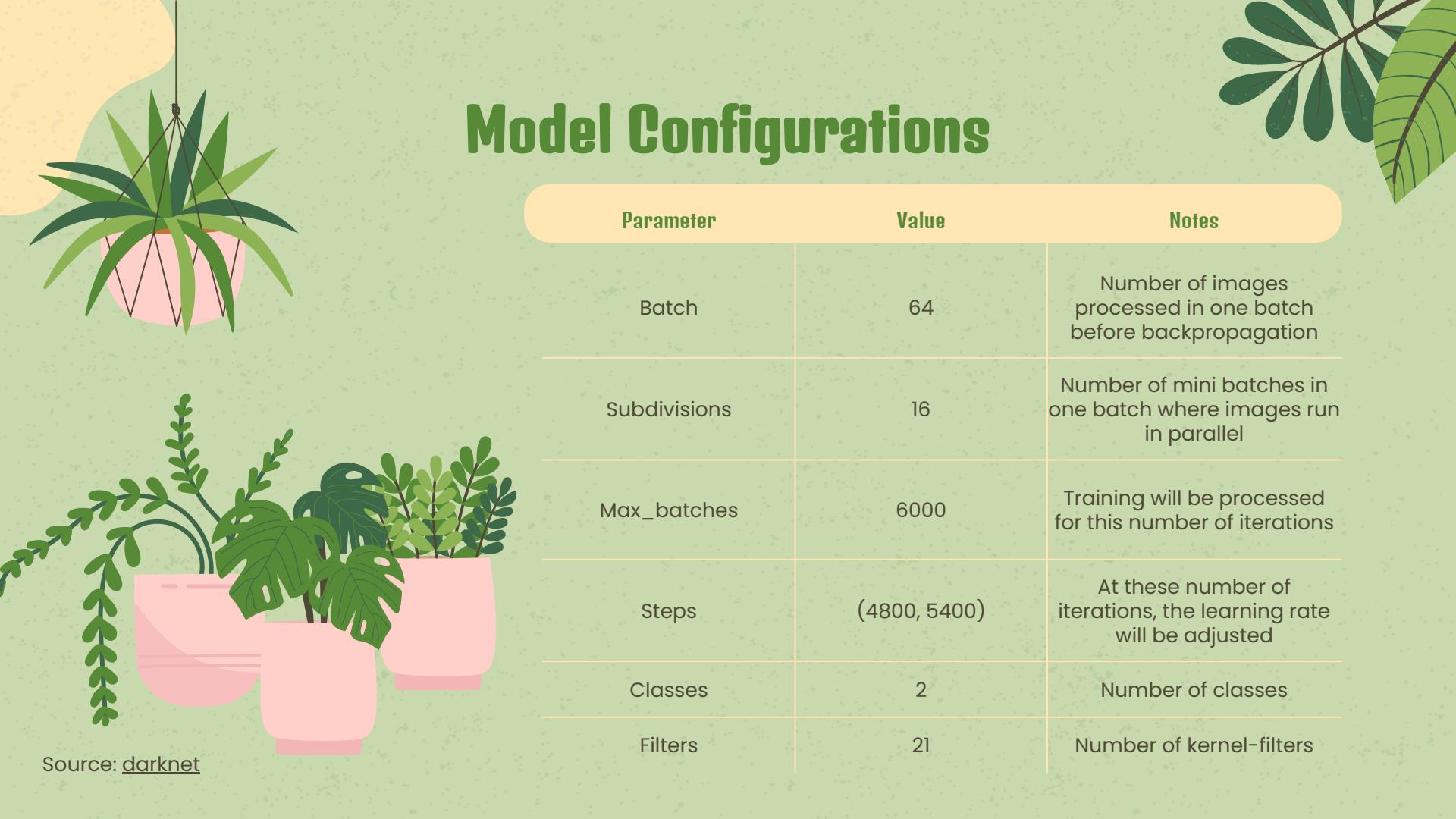
Unhealthy

Crop leaves (278)

Houseplants (440)

Reddit (310)

TOTAL: 1028

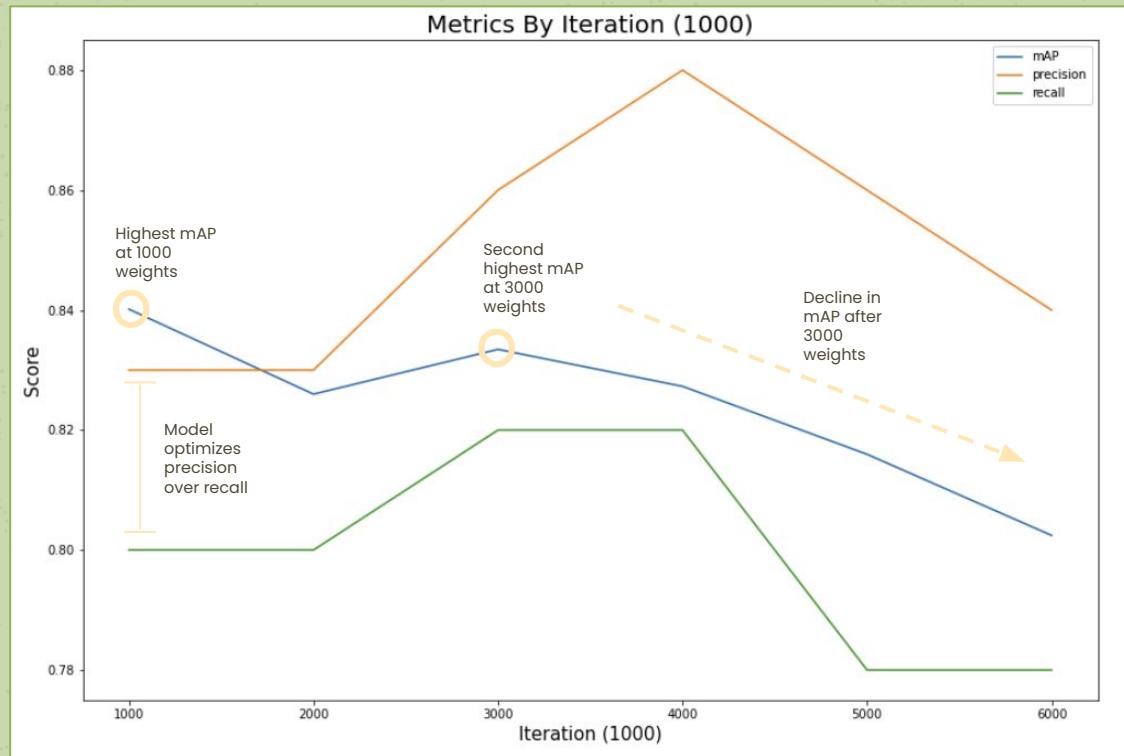


Model Configurations

Parameter	Value	Notes
Batch	64	Number of images processed in one batch before backpropagation
Subdivisions	16	Number of mini batches in one batch where images run in parallel
Max_batches	6000	Training will be processed for this number of iterations
Steps	(4800, 5400)	At these number of iterations, the learning rate will be adjusted
Classes	2	Number of classes
Filters	21	Number of kernel-filters

Source: [darknet](#)

YOLO Model Performance





04

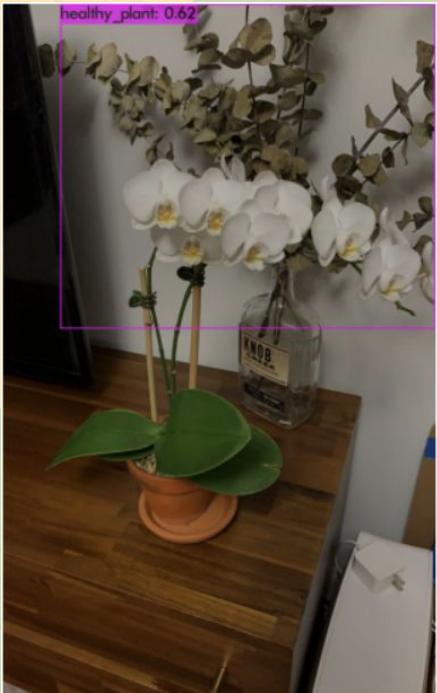
Demo

Let's detect some plants!

High Confidence + Accuracy



Ambiguous Cases



Confusion



A vibrant, stylized illustration of tropical foliage. It features large green leaves with prominent veins, some with serrated edges. In the bottom left corner, there are pink flowers with long stamens. The background is a textured light green.

05 Conclusion & Next Steps

Conclusion

- 1 **For the YOLO model, mAP fluctuated around low 80%.** Typically with YOLO models, mAPs increase significantly in the first couple thousand iterations and then plateaus in the next few thousand iterations. Hitting upper limit makes sense given that it can sometimes be difficult to distinguish unhealthy plants from healthy ones, even for humans.
- 2 **A major reason for model confusion is due to the huge variety in plants.** An unhealthy characteristic of one plant might not be for another. For example, some plants' leaves naturally grow downwards but it could be mistaken as wilted and therefore classified as unhealthy by the model.
- 3 **The CNN model capped around 80% accuracy which is also likely due to the confusion mentioned above.** While it seems that YOLO did not have a huge improvement over CNN, the benefit of YOLO is the ability to detect in real-time. For demonstration purposes, the 3000 weights iteration was used as it gives us high precision with additional training as opposed to the 1000 weights.



Next Steps

TRAIN ON ADDITIONAL IMAGES

As mentioned, the model may confuse a normal characteristic of one plant to be unhealthy on another plant and vice versa. The easiest way to reduce the confusion is to train the model on a wider variety and more images of healthy and unhealthy plants. The immediate next step could be to scrape more images from r/plantclinic. Additionally, the "real-life" images available on the subreddit would help the model learn better on images that it can expect to receive in real-time.



CREATE MOBILE APP

Once the model is optimized, it can be used to create a mobile app where users can detect whether their plant is healthy or unhealthy. Users can either use their phone cameras to detect plant health on the spot or set up a webcam and view detections on their phone while they are away. This can be used by people who travel often but still want the option to check in on their plants.



THANK YOU!

Any Questions?





Special Credits

GA instructors Chuck, Grant, and Varun for helping me navigate the intricacies of YOLO.

GA classmates Christina, Rachel, Shashank, Stephen, and Trevor for providing me pictures of their plants for the demo.

Thank you!

CREDITS: This presentation template was created by **Slidesgo**, including icons by **Flaticon**, and infographics & images by **Freepik**