What Kind of **Sick Tomato** are You?

Yixi Yang,
John Scott,
Patricia Gallagher

W207 Summer 2022

Why do we care about sick tomatoes?

- For amateur gardeners, it is very hard to know why your garden veggies are sick
- Makes purchasing cures difficult or even dangerous
- Allowing home gardeners to easily identify disease type can save harvests
- We had 9 categories of tomato diseases and a healthy class amongst our 38 plant diseases
- Chose tomatoes for most interesting image classification problem

Dataset Description

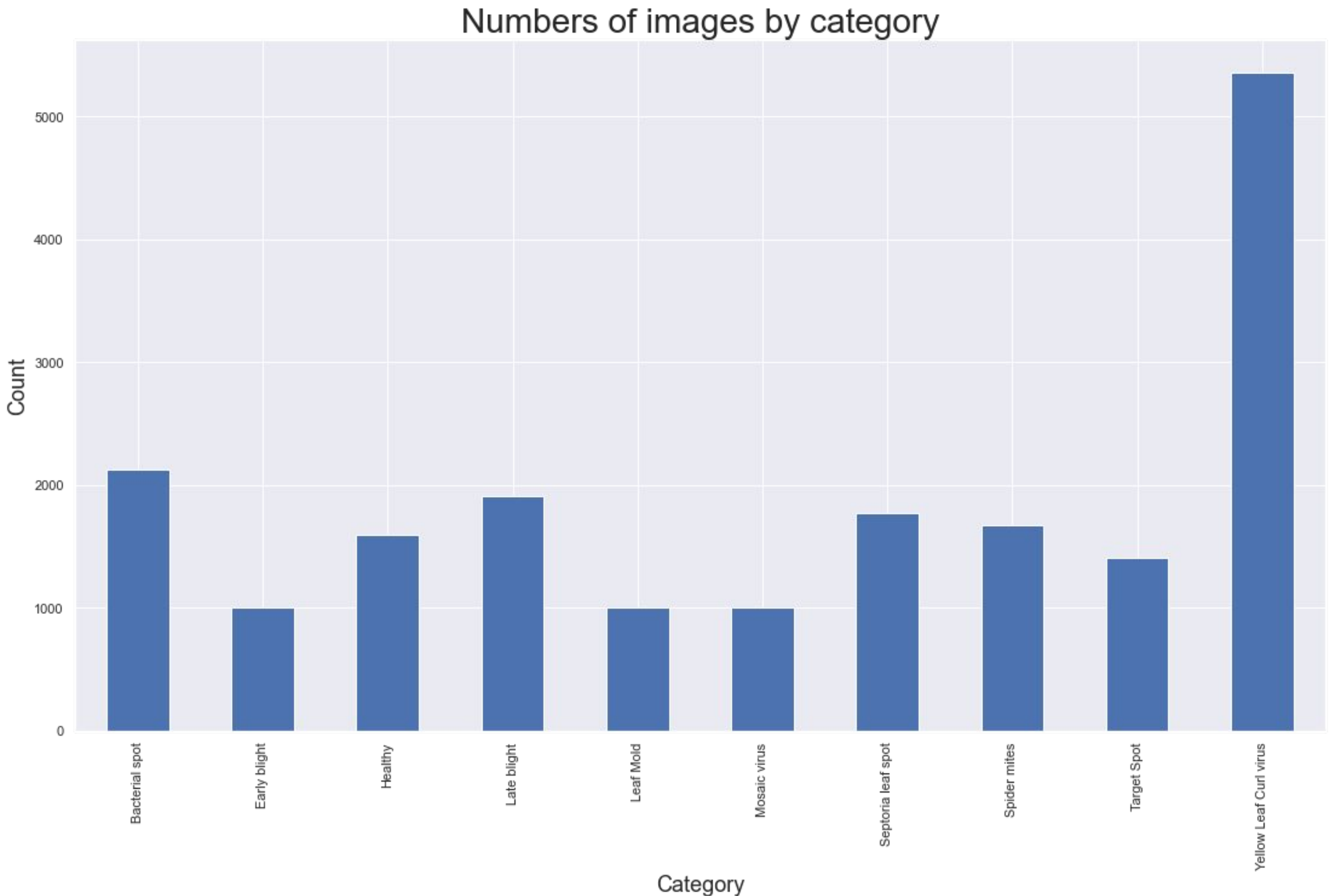
We carved out over 18,000 images of tomato plant leaves (256 x 256) from a much larger plant leaves diseases image database, obtained from paper in *Mendeley Data*

The tomato leaves fall into 9 disease classes and 1 healthy class

We split the data into 70/10/20 for training/validation/testing

It is an unbalanced dataset with different number of images per class

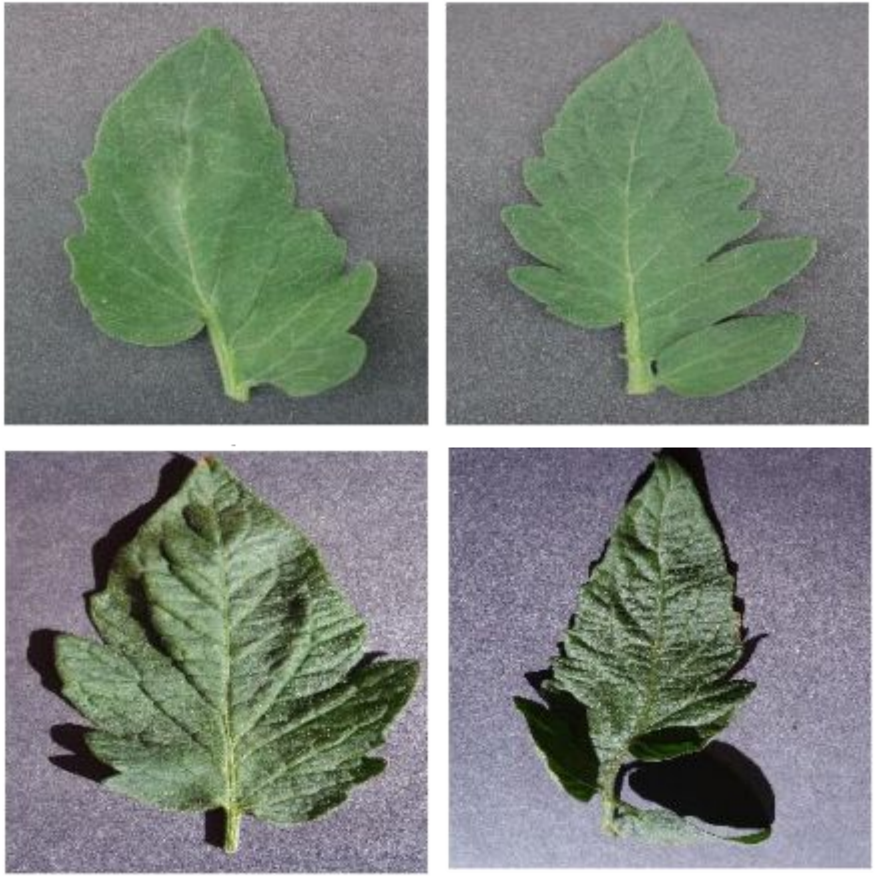
Bacterial Spot	Early Blight	Healthy	Late Blight	Leaf Mold	Septoria	Spider Mites	Target Spot	Mosaic Virus	Yellow Leaf Curl
2127	1000	1591	1908	1000	1771	1676	1404	1000	5357


Numbers of images by category
Count
0
1000
2000
3000
4000
5000
Bacterial spot
Early blight
Healthy
Late blight
Leaf Mold
Mosaic virus
Septoria leaf spot
Spider mites
Target Spot
Yellow Leaf Curl virus
Category

Healthy Tomato Plants

Differences between healthy plants

1. Background
2. Lighting (flash vs flat lighting)
3. Shadows from wrinkles and leaf
4. Ripped edges vs continuous edges
5. Leaf shape
6. Color



Yellow Leaf Curl Virus

Early Blight

Bacterial Spots

Late Blight

Leaf Mold

Septoria

Target Spot

Spider Mite

Tomato Mosaic Virus

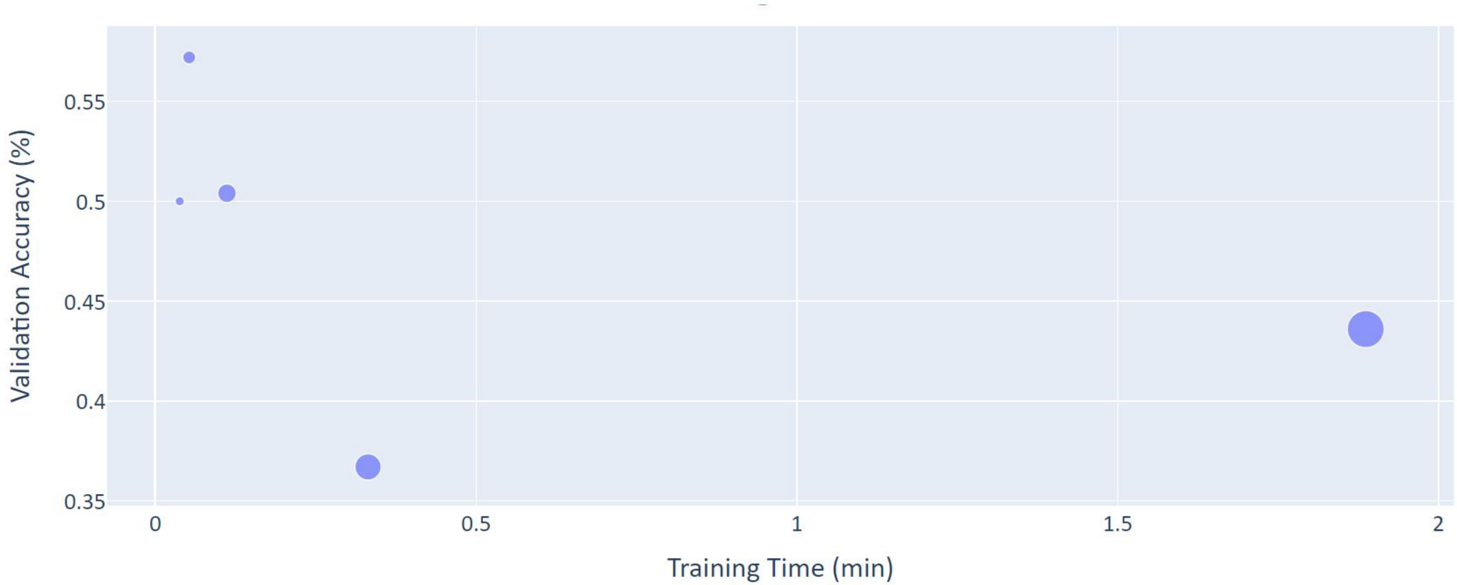
HOOMANS

Solution approach: models of increasing complexity

1. **Logistic classification**
   a. Different image sizes
2. **Feed-forward neural network**
   a. (1, 2, 3, or 4 layers deep)
3. **Simple CNN**s with multiple layers
   a. 1-2 convolutional layers, 1-2 dense layers
4. **Alex-Net CNN**
   a. Imitated structure, trained from scratch
5. **Transfer Learning**
   a. VGG16, VGG19 using ImageNet weights
6. **Ensemble model**
   a. Combine five high-accuracy, quickly training neural networks from above

Experiments: Logistic results

- Validation accuracy peaked around 57%
- Image size is the only varied hyperparameter: 32x32 great, 16x16 overfitting
- Larger image size produced much less accurate results



Experiments: FFNN results

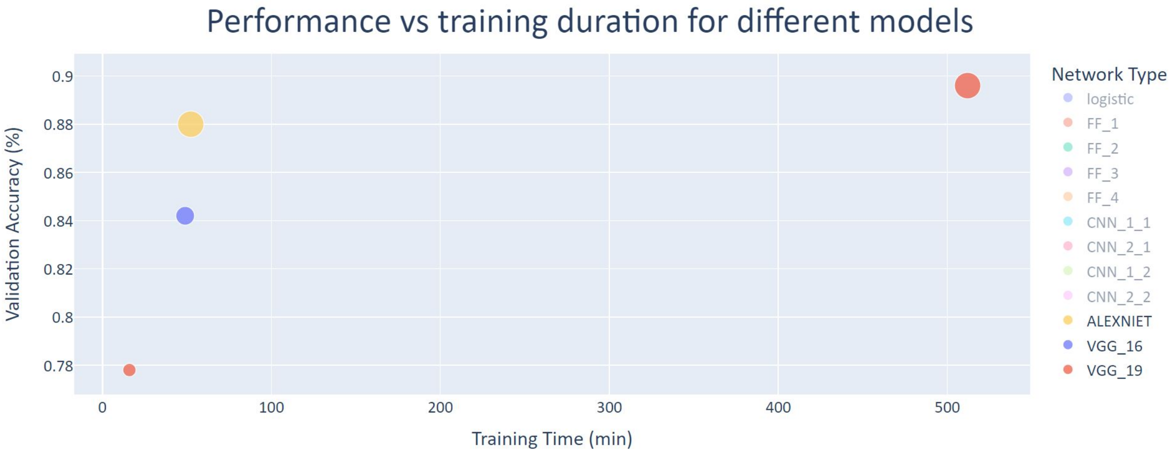
- Validation accuracy peaked around 70%
- Batch size important
  - Enough to sample image
  - Bigger = faster to run
- # of layers and image size are relatively unimportant
- # of nodes in dense layers, especially the last dense layer, is essential
- Given same # of nodes in the final layer, adding dense layers does seem to marginally improve validation accuracy

Experiments: CNN models with varying parameters

- Experimented many different NN structures, with variety in
  - Kernel size - runtimes improved dramatically after increasing kernel size
  - Number of convolution and pooling layers
  - Number of dense layers
- Almost all CNN models produced validation accuracy over 80%

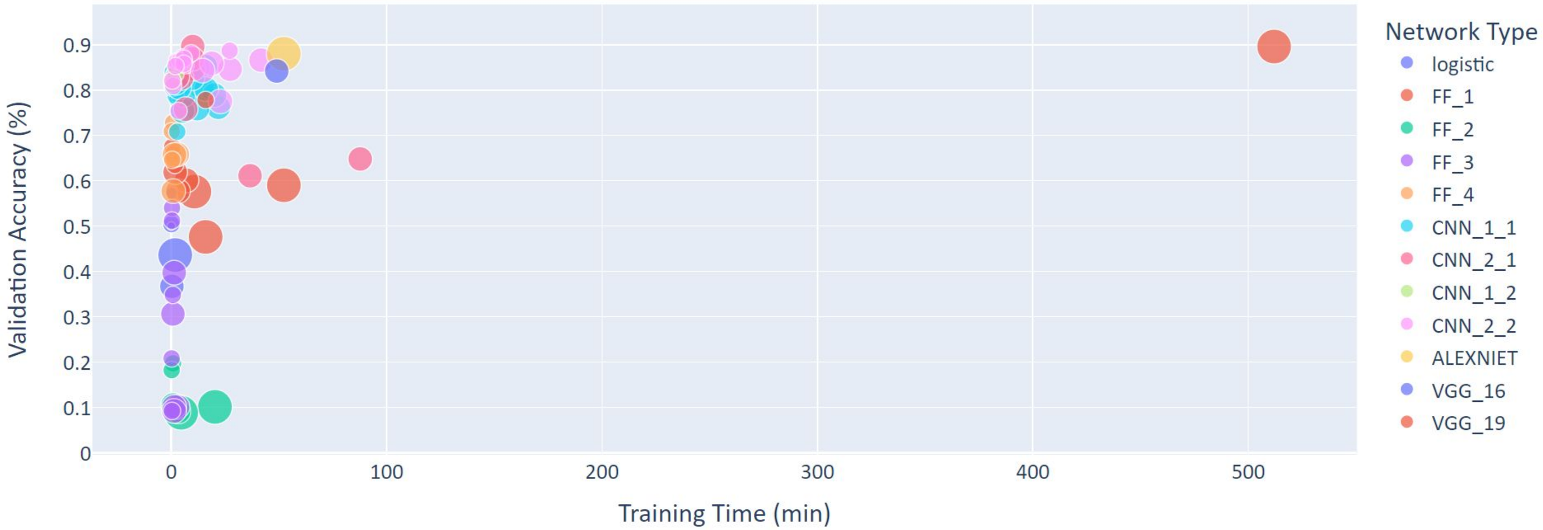
Experiments: CNN models with transfer learning

- Handbuilt a model based on AlexNet - we call it AlexNiet
- Used transfer learning using models VGG16 and VGG19 from keras
- VGG16 and VGG19 were built upon AlexNet, the CNN that revolutionized image classification
- We added layers to VGG16 and VGG19 such as extra dense layers



Conclusion: CNNs are best, even simple ones

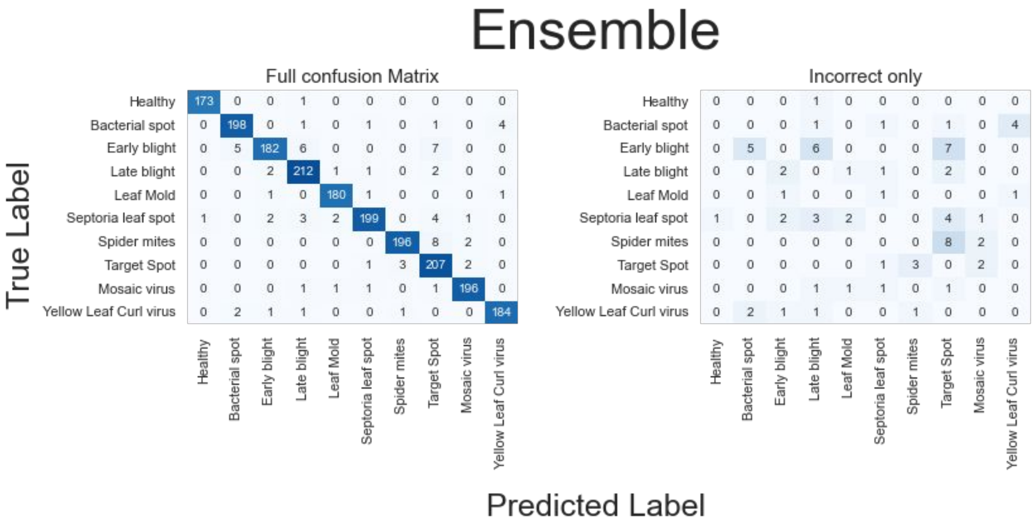
- CNNs overwhelmingly did better than logistic models and FFNN models
- Highest validation accuracy 90%, tie between:
  - Transfer learning from VGG19 + 2 dense layers (1024, 256)
  - CNN with 2 convolutional layers and one dense layer



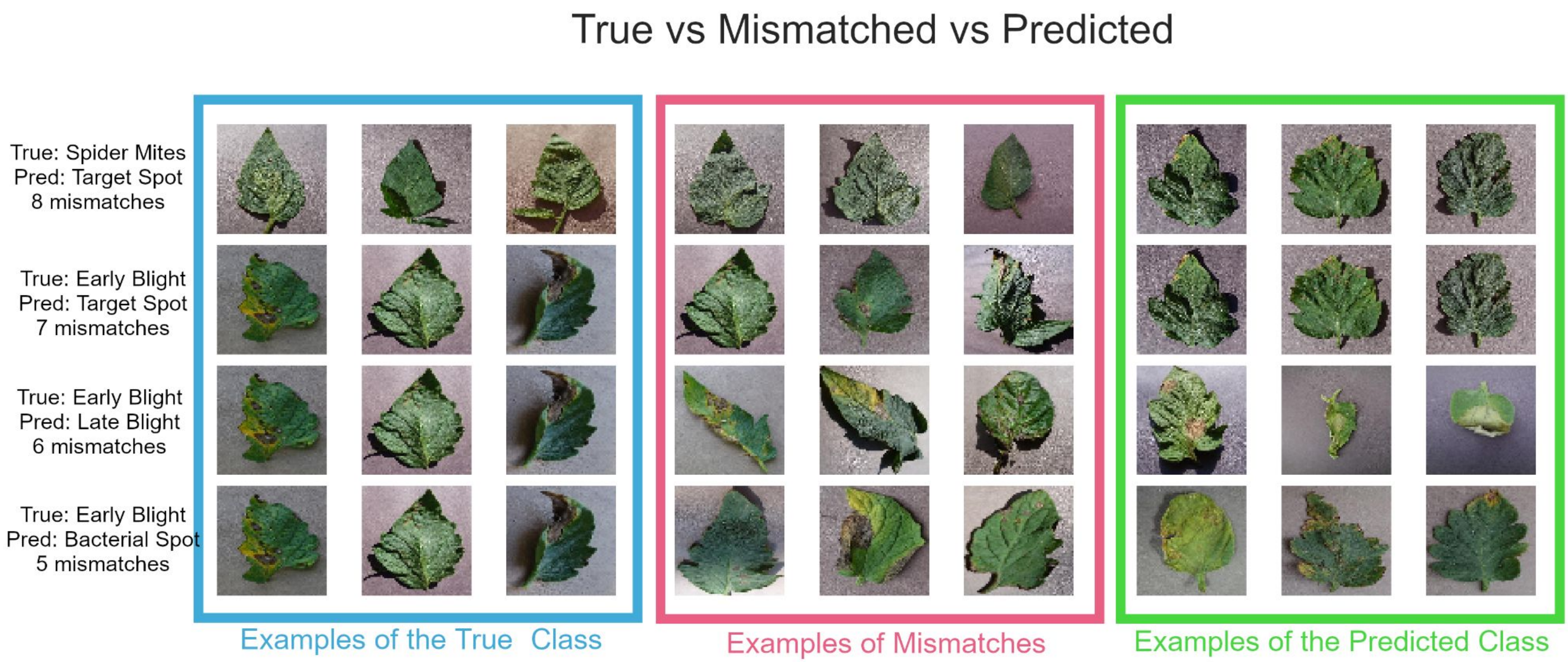
Performance vs training duration for different models
Validation Accuracy (%)
0.1
0.2
0.3
0.4
0.5
0.6
0.7
0.8
0.9
Training Time (min)
0
10
20
30
40
50
60
Network Type
logistic
FF_1
FF_2
FF_3
FF_4
CNN_1_1
CNN_2_1
CNN_1_2
CNN_2_2
ALEXNIET
VGG_16
VGG_19

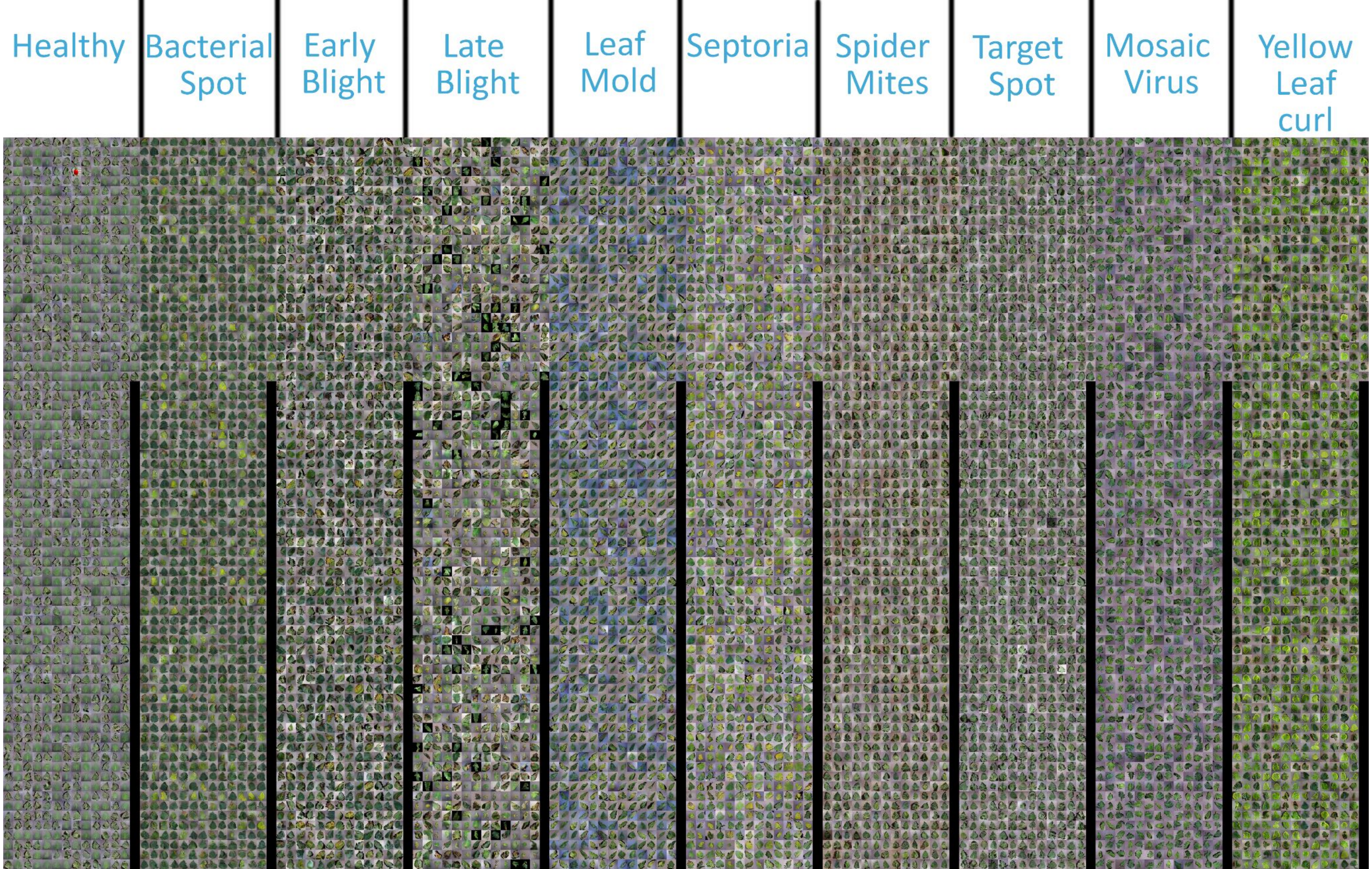
Conclusion: Ensemble is Better

- We took the average weights trained from five high-accuracy, high-speed CNNs
- Resulting confusion matrix shows fewer misclassifications



Examples of Confusion




Healthy
Bacterial Spot
Early Blight
Late Blight
Leaf Mold
Septoria
Spider Mites
Target Spot
Mosaic Virus
Yellow Leaf curl

Conclusion: important hyperparameters

- Ensure relatively large set of training images, with at least 1,000 per class
- Less is more (or nearly as good as) for number of layers
- More is more for batch size
  - Small batch size lengthened runtime unbearably
- Less is fast for image size
  - 256x256 can take multiple times longer than 128x128
  - 64x64 is extremely fast while validation accuracy is comparable
- For CNNs, kernel size is essential
  - Too small slows down calculation and fails to capture image well
  - Validation accuracy and runtime improved drastically just by increasing kernel size from 2 to 5
- The ensemble performed the best in terms of training time and accuracy

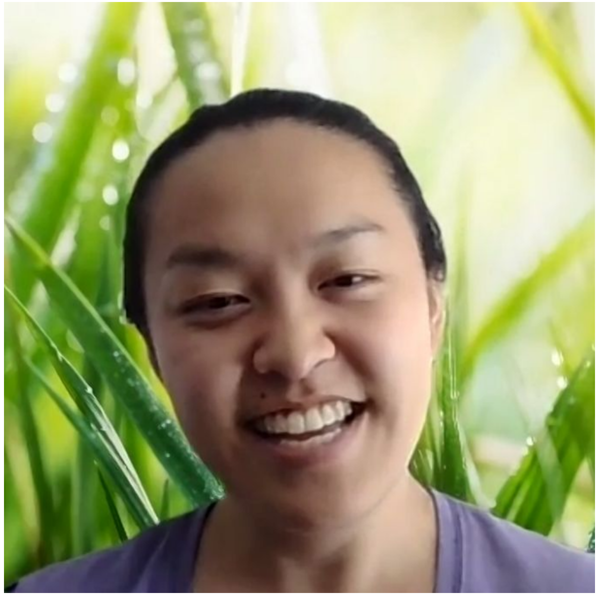
Potential Negative Societal Impacts

- Relatively low risk: we assess net impact to be **positive**
- Methodology: no human-derived data
- Application: real-world model application does not directly impact human access to resources or treatment of human beings
- If misused: could treat the wrong disease and tomato remains sick
- Potential positive societal impacts
    - **Democratizes** access to plant care and disease identification
    - Makes at-home gardening easier
    - Encourages local food consumption
    - Alleviates food deserts
    - Reduces global carbon footprint

Next Steps

- Modeling changes
  - Image **segmentation** prior to classification to remove background
  - Data **augmentation** to simulate different lighting conditions
- Expand to **additional plants**/diseases
- Measure **desirability**
  - Freshness / edibility
  - Beauty pageant (*#1 most beautiful tomato in the MidWest*)
  - Pricing (based on quality rather than looks)
- Expand to **other parts** of the plant (other than leaves)
- Mushrooms, **legumes**, nuts, etc
- Human **diseases** (moles / rashes / etc)

Just for Fun - Apply our Model to Human models!



Late Blight

Late Blight

Late Blight

To Do

- Update the html plotly chart - done
- Tiled summary of images
- GitHub repo with our code
- Final submission
    - PDF of the report
    - Slide Deck
    - GitHub link (with key resources there)
    - HTML file

Edit the reports!

530 pm PDT tuesday! Before class!