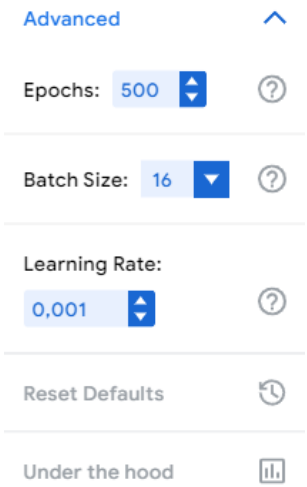
Machine learning tests with Google Teachable Machine

Teachable machine is a Google technology which uses TensorFlow.js library. Under the hood they use a technique called transfer learning. This technique focuses on storing knowledge gained while solving one problem and applying it to a different but related problem. To use Teachable Machine all we have to do is select the image learning model, put in our different classes and load our image data. After this we can configure the training model as follows.



Epochs: One epoch means that each and every sample in the training dataset has been fed through the training model at least once. If it is set to 50 it means that the model we are training will work through the entire training dataset 50 times.

Batch Size: A batch is a set of samples used in one iteration of training. Result of Total image count / Batch size gives us the total batches. And once all 5 batches have been fed through the model, one epoch will be complete.

Learning Rate: Does not have a clear definition in Teachable Machine but it has been known that it controls how quickly the model is adapted to the problem.

Rice Diseases Image Dataset: I have made a couple of tests using Teachable Machine with different parameters. All of the test have been done with:

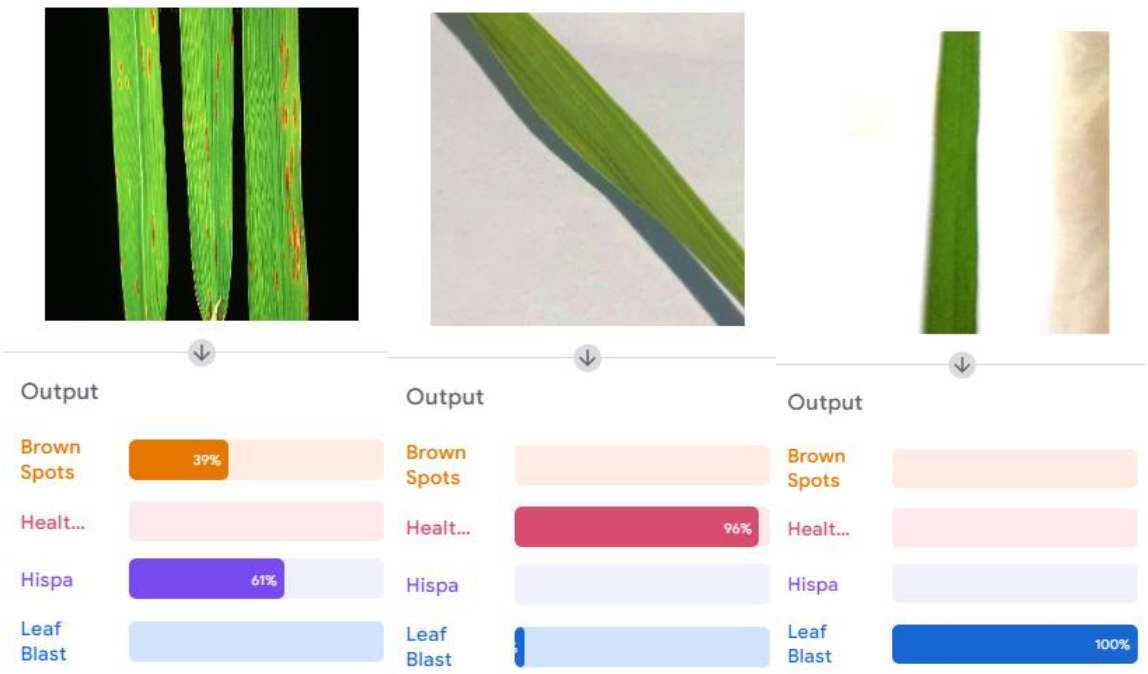
523 Images of Brown spots, 1488 Images of Healthy

565 Images of Hispa and 779 Images of Leaf Blast classes

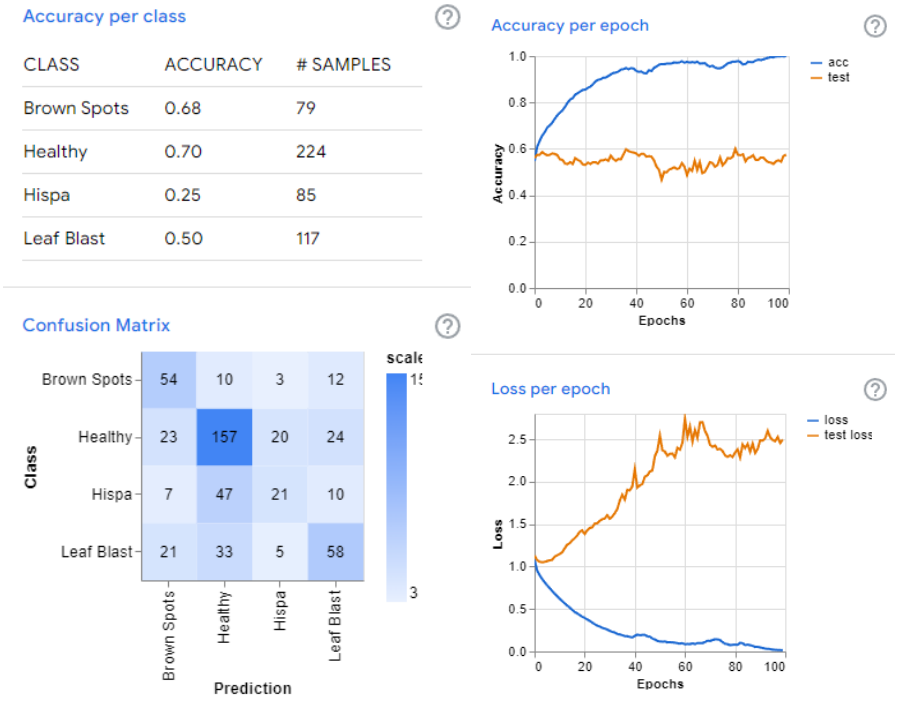
Test 1: 50 Epochs, 16 Batch Size, 0,001 Learning Rate.

Confusion Matrix: Teachable machine automatically chooses some of the images from our training file as test images and doesn't show these images to the model. As we can see from one of the screenshot the #SAMPLES column contains how many of these images were selected as test(validation) images. Then by itself the model runs a test on these images. Confusion Matrix shows us how many of the images have been found correctly and how many of them were found wrong.

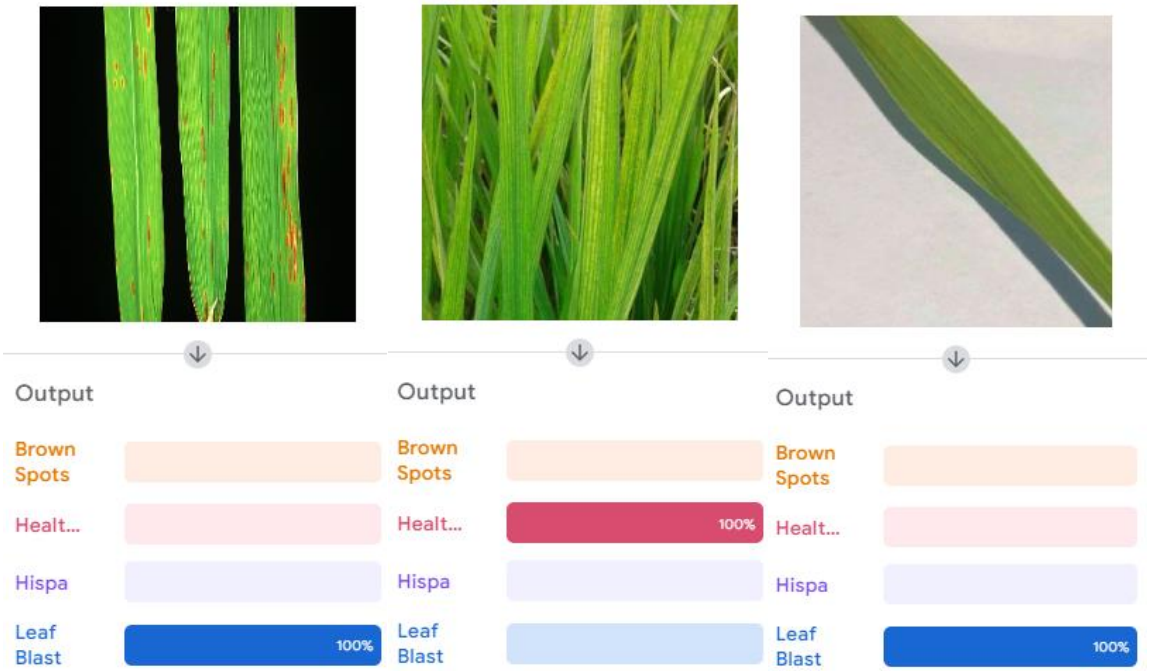
Although accuracy per class is high, this model fails some of the tests. It struggles to be sure about Hispa class the most and it choses Leaf Blast class in most cases. Except for these issues it seems to be working fine.



Test 2: 100 Epochs, 16 Batch Size, 0,001 Learning Rate.



When we increase epochs to 100 we see that accuracy per class reduces. But we see that in some examples it does the job better than 50 epochs. Still this model won't be suitable for a big project.

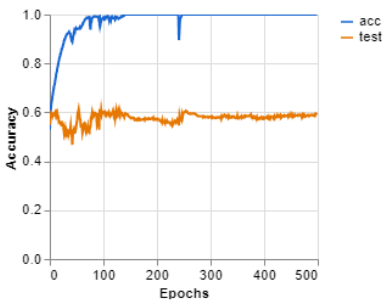


Test 3: 500 epochs, 16 Batch Size 0,001 learning rate

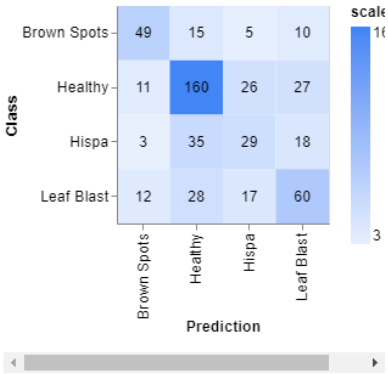
Accuracy per class

CLASS	ACCURACY	# SAMPLES
Brown Spots	0.62	79
Healthy	0.71	224
Hispa	0.34	85
Leaf Blast	0.51	117

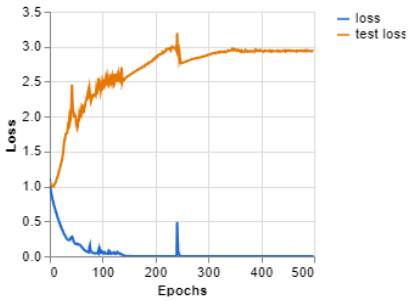
Accuracy per epoch



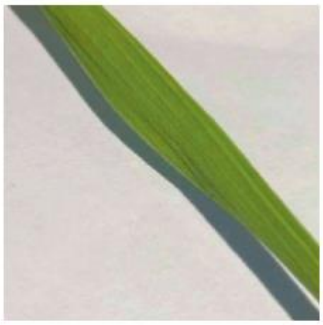
Confusion Matrix



Loss per epoch

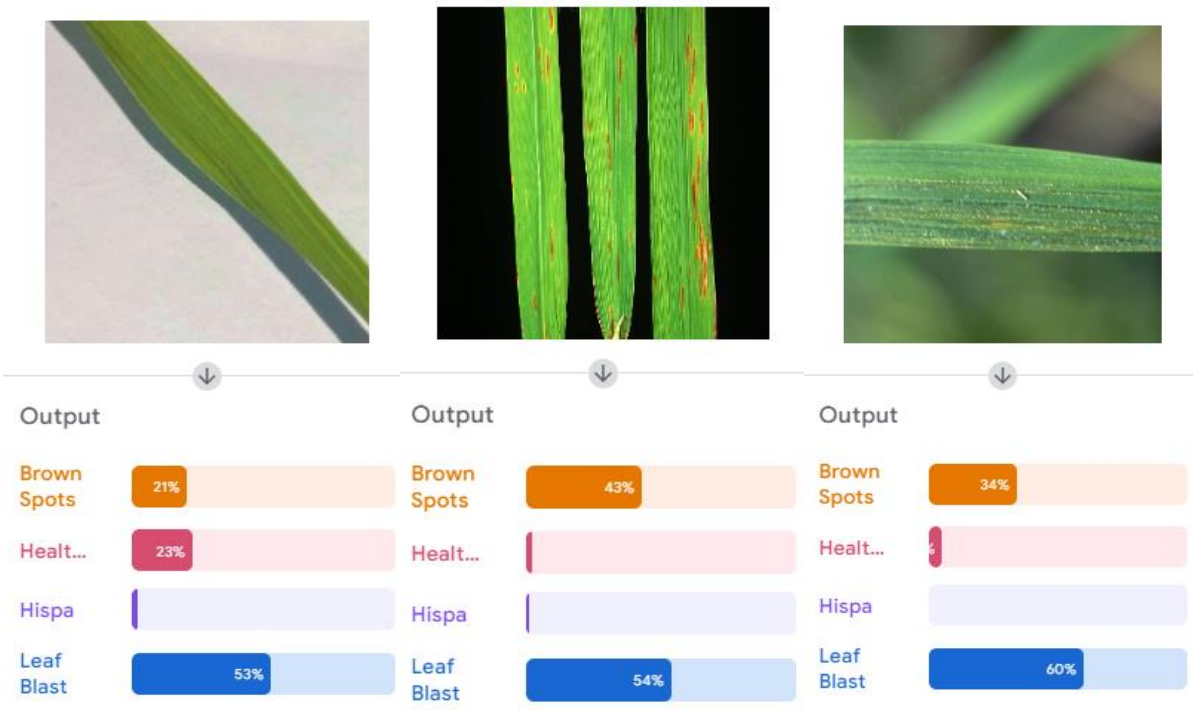


When we increase epochs to 500 we lose some accuracy for some classes and increase accuracy for Hispa class. But overall many tests fail regardless of high epoch number.



Test 4: 50 Epochs, 16 Batch Size, 0,00001 Learning Rate

Even though we see that accuracy increases when we lower the learning rate. In the tests we can also see that it fails more often than higher learning rate models.



Conclusion: Teachable Machine from Google is definitely a great tool to use. It can still be used in projects that don't require a strict selection or that have at most 2 classes. The main problem with getting lower accuracy from these tests may be from some faulty data within the dataset. If we organise the dataset further we may increase our chance of getting better results.