Training your models

What's a model?

Just like a lab assistant can learn about the features of control and mutant images and then tell if an image is control or mutant, a machine learning model can classify images by learning from relevant data. In other words, the model tries to relate the given images with their labels, just as the human brain does. Essentially machine learning model is an algorithm which learns features from given data to produce prediction labels later, i.e., “control” or “mutant”, “dog” or “cat”.

Fig. machine learning service for image classification which takes images and produces labels for them.

What's training a model?

Before making predictions on your images, you’ll start by training a model in the web app so that it learns about the features. For example, to train a model to become a cats and dogs classifier, we’ll give the application lots of pictures of cats and dogs. It then “learns” the characteristics of the images that determine which are of cats and which are of dogs. We call the results of this a trained model.

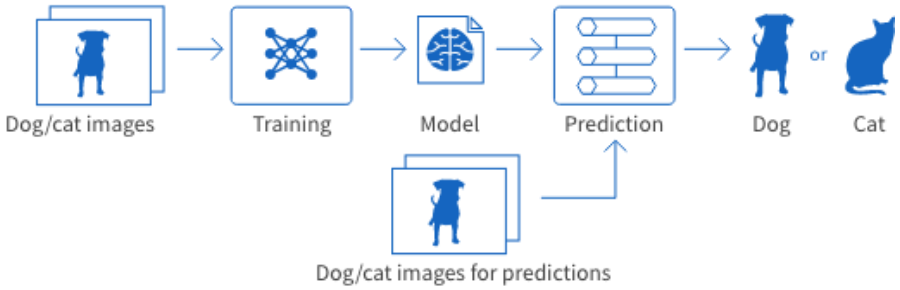


Fig. Train a model with images of cats and dogs, and test using this model to classify new images.

This process can be generalized to perform any similar tasks, for example instead of dog/cat, the model can be trained to classify control/mutant frog embryo images.

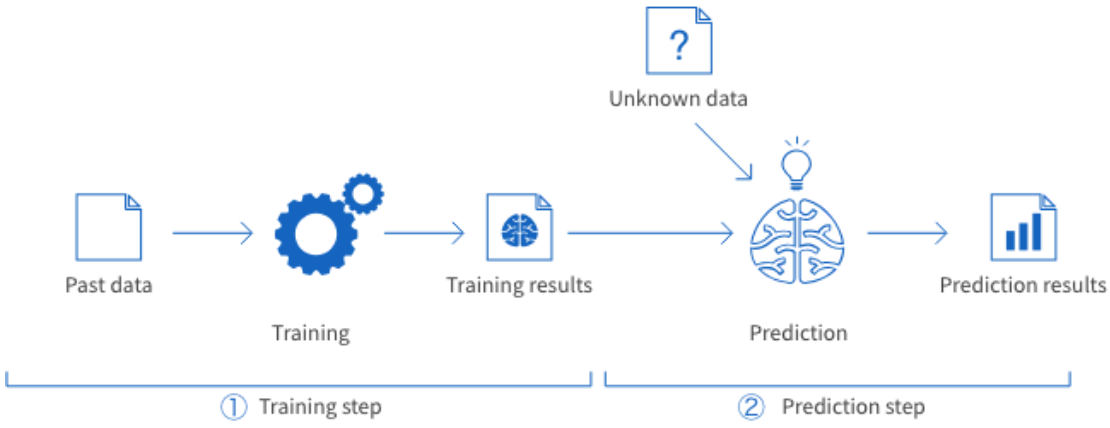


Fig. a generalized training and prediction workflow

In this web application, the steps to train a model include:

- Collect and organize data into categories.
- Create a project, define categories of the project, and upload data to each category.
- Customize hyperparameters (optional).
- Name and train your model.
- View model evaluation once it's finished training.

Step 1. Data preparation.
Before starting using the web application, select the subset of all available data that you will be working with. Organize your data into a train folder and test folder, which contains subfolders of each category you want the model to learn.

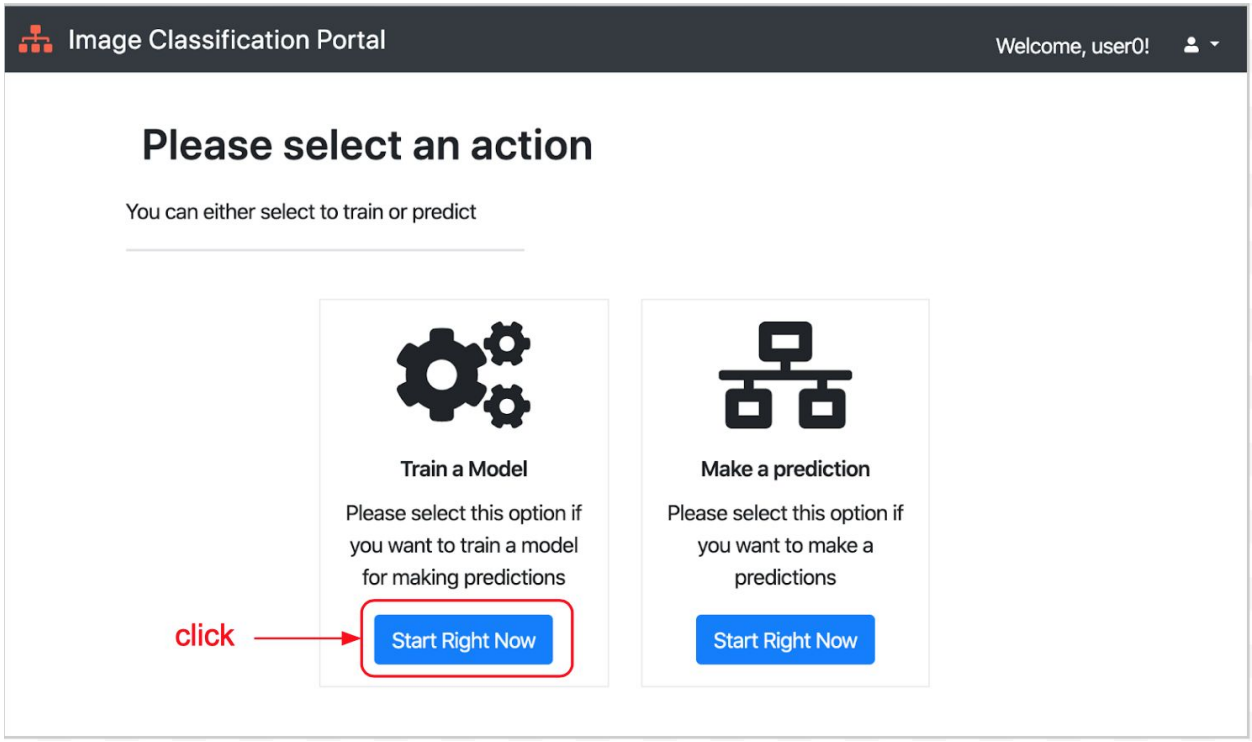
For example a training data directory and testing data directory containing one subdirectory per image class, filled with.jpg images:

The more images you have in your training data, the better the model will learn. Images in testing data are used to evaluate your model’s performance after it’s trained. Minimum 100 images in each category for training is recommended.

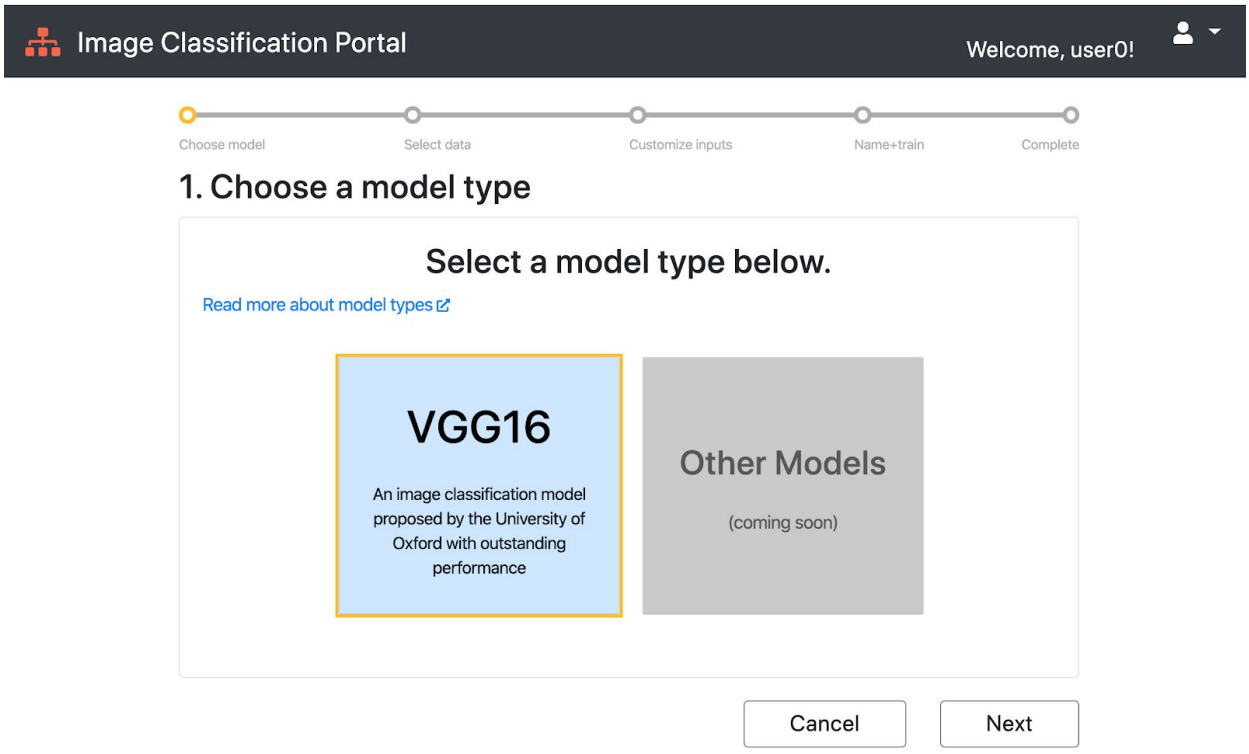
- The images must be in JPEG formats.
- Each image must be 30MB or smaller. Note that there's generally no accuracy benefit to providing very high resolution images.
- Include at least 10, and preferably 100 or more, examples of each label.

Step 1
Log in or sign up

Step 2
Select “Train a Model”



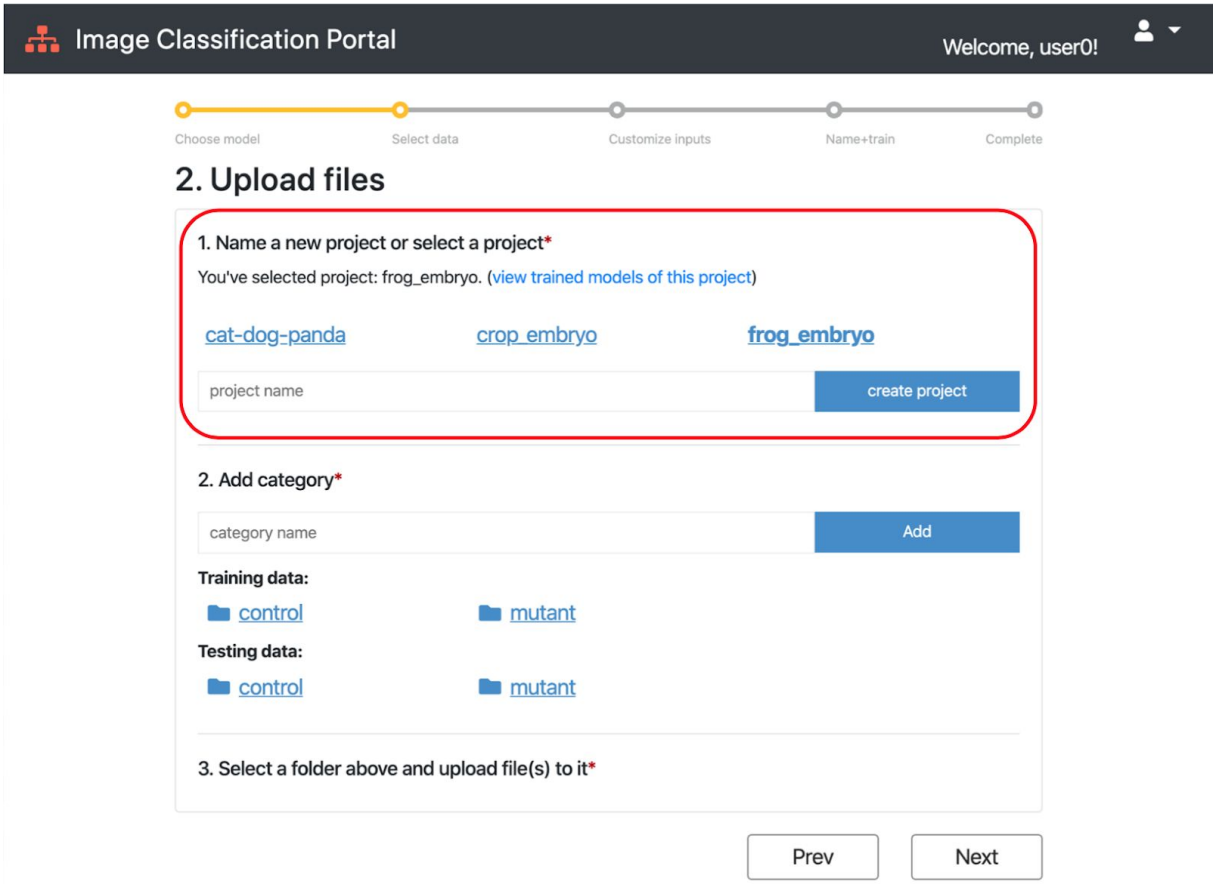
Step 3
Select a base model type and got to “Next” (currently only VGG16 model available)



Step 4

Select an existing project or create a project.

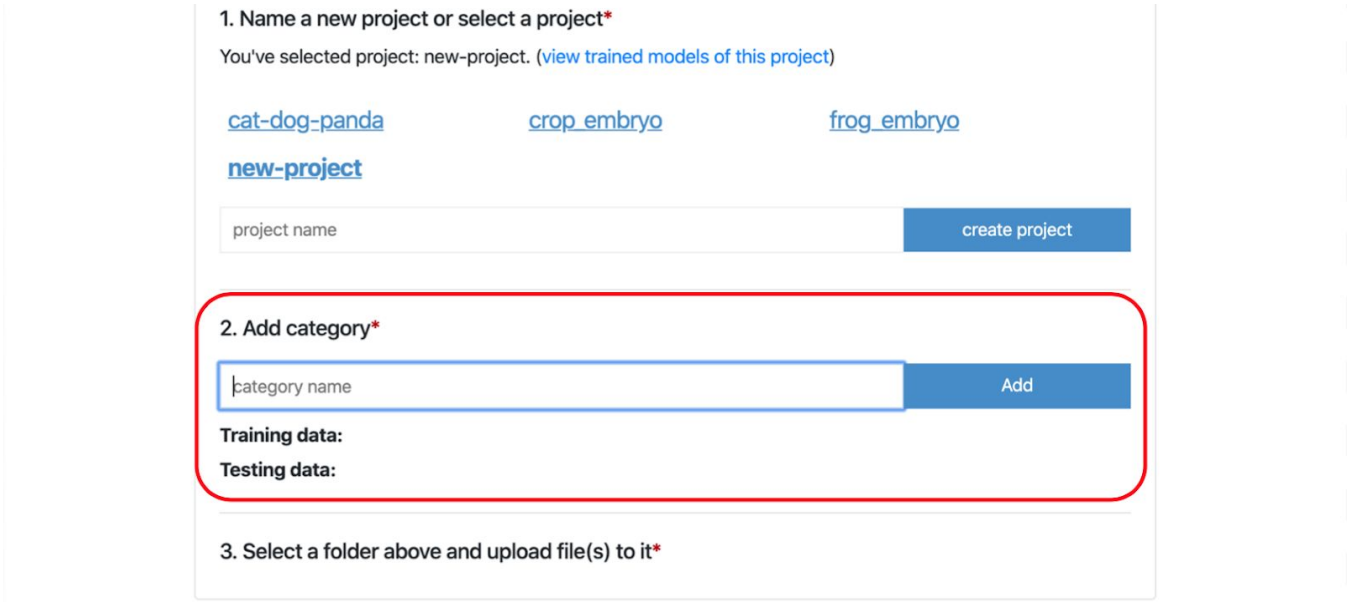
The project is where your training and testing images exist. You can train multiple models on any project with variations of parameters (next step).



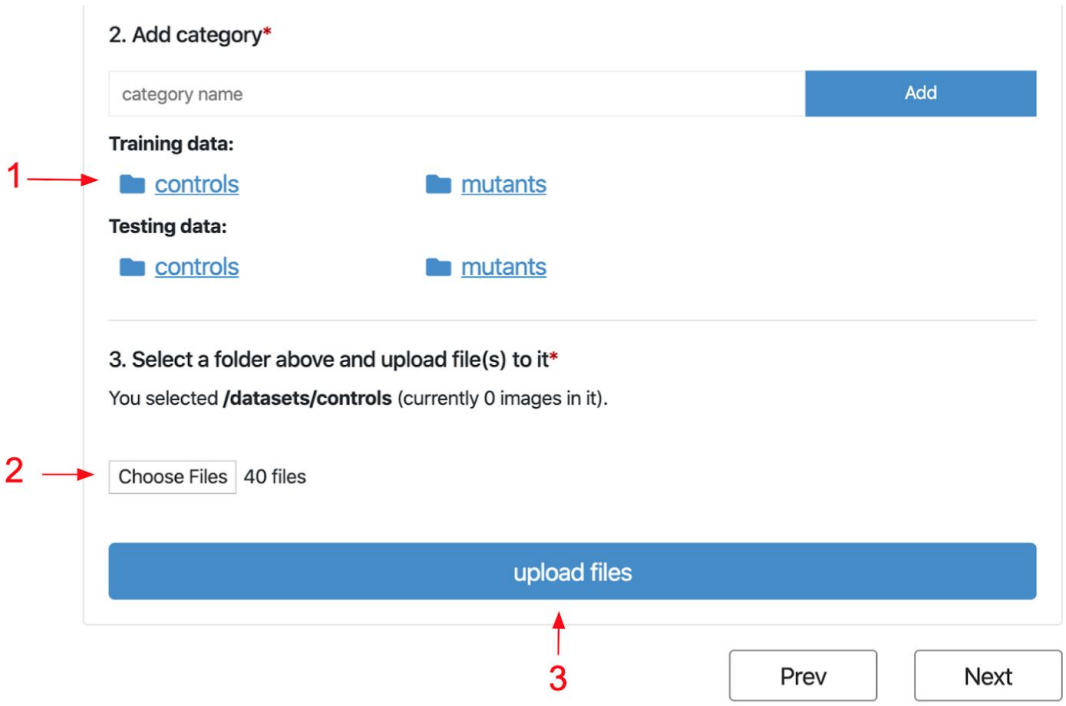
Step 5

Add categories (labels) in the project you selected as needed, e.g. “controls” and “mutants”.

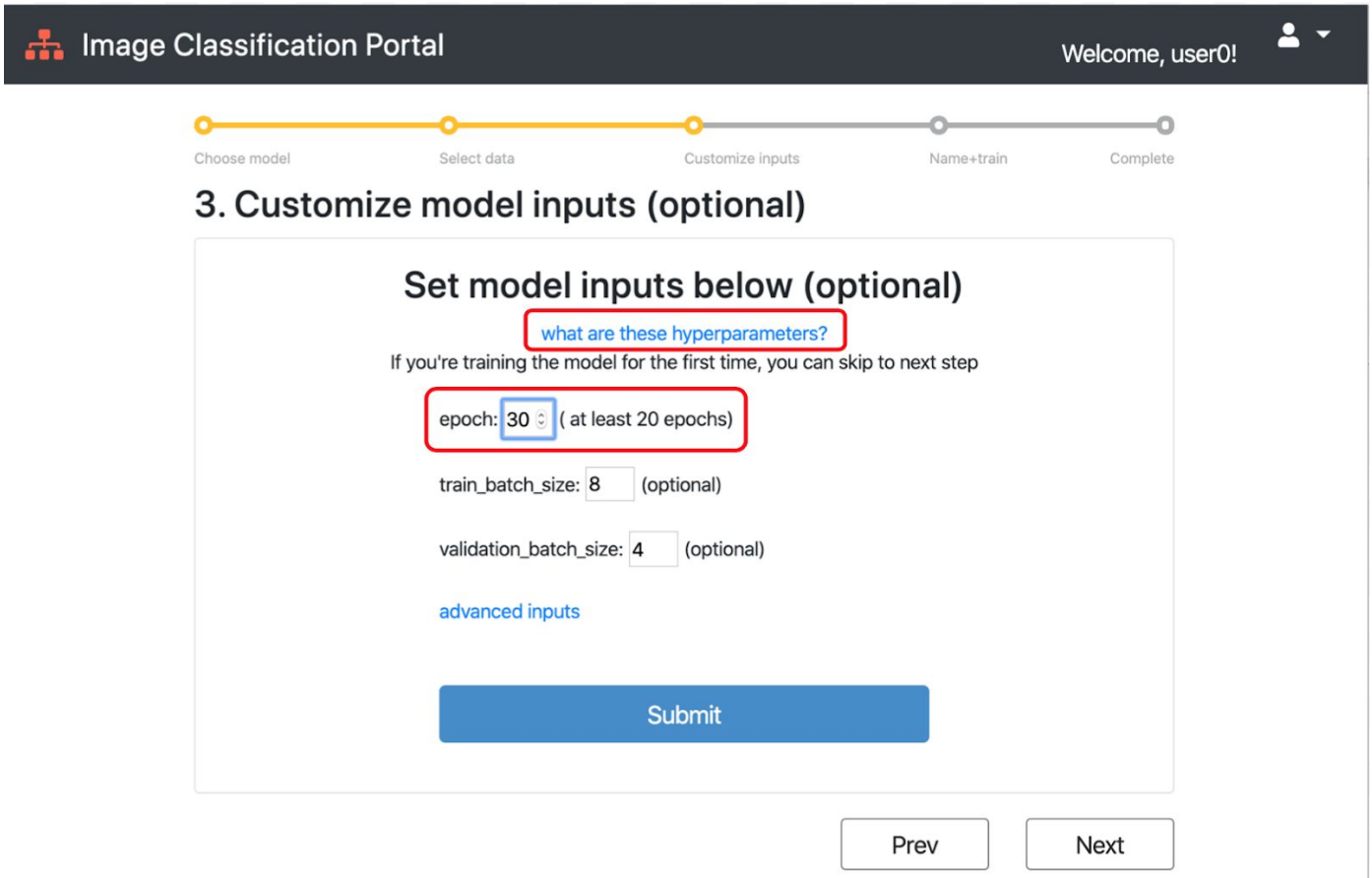
Note: the category is created in both ”Training data” and” Testing data” once you click “Add”.



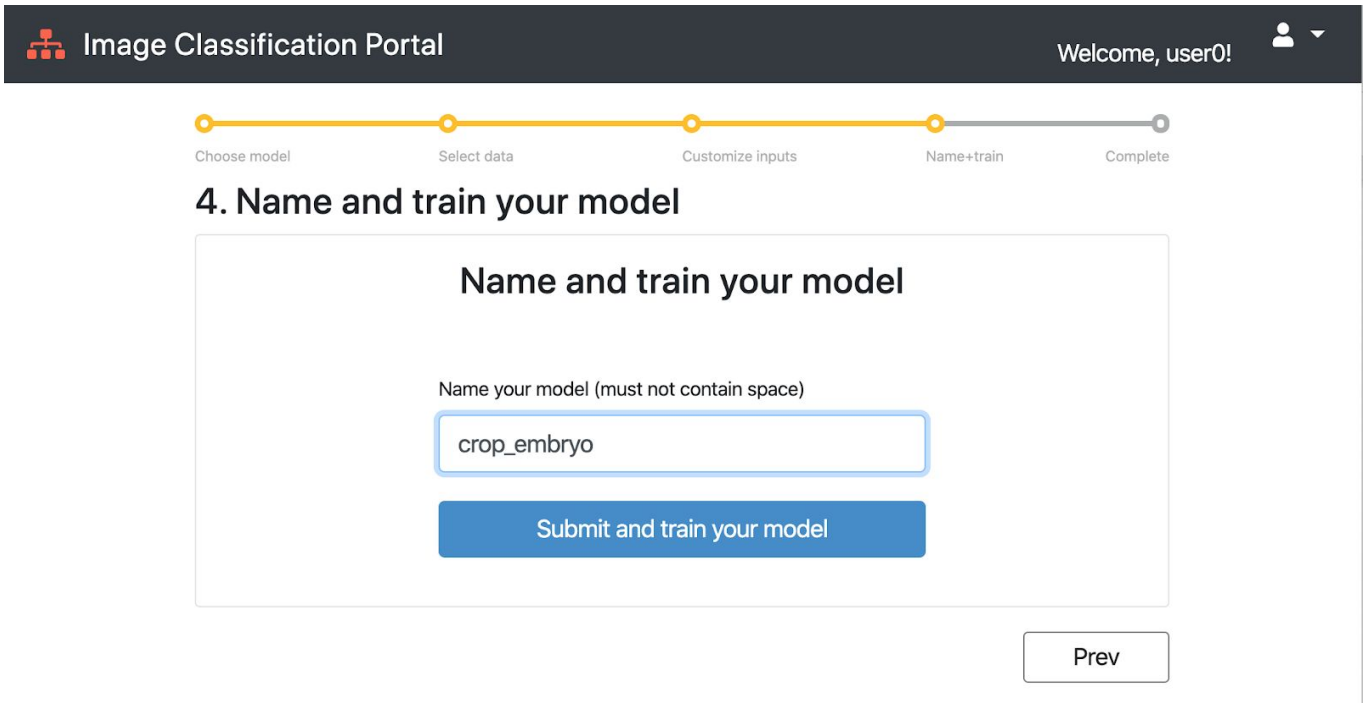
Step 6
Select each folder from training data or testing data and upload images.
Try to add as many images as possible to each folder in training data.
Testing data is used to evaluate your model after it's trained with training data.
After the images are uploaded you can scroll down to see their preview.



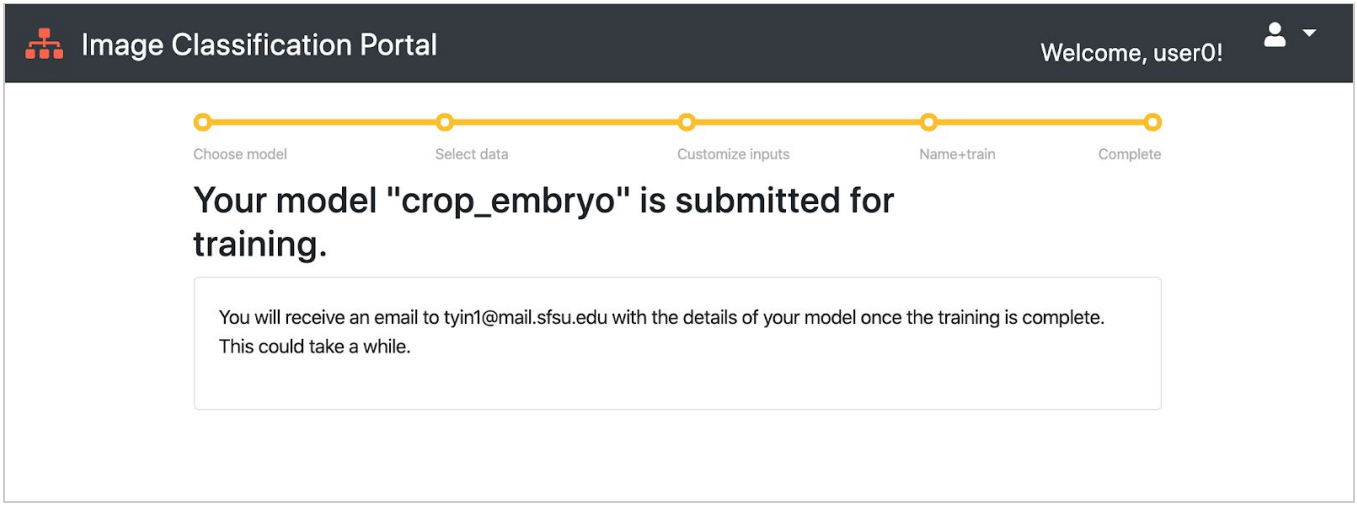
Step 7 (**optional**)
Read about hyperparameters with the link provided.
Set your own parameters, click submit and click next.



Step 8
Type a name for your model and click “submit and train your model”.



Step 9
Wait for the model to complete training.
This could take a while depending on your data size and number of epochs.



Step 10

Once your model finishes training, you'll receive an email with training results. Click the link to view details. View training results, read the suggestions, and decide your next action.

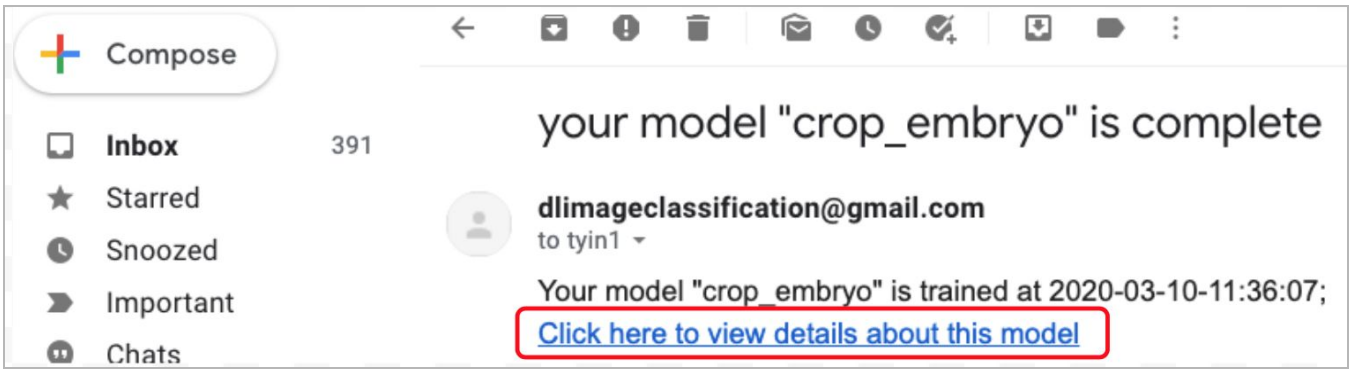
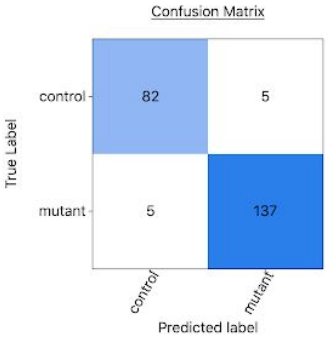


Image Classification Portal Welcome, user0!

Choose model — Select data — Customize inputs — Name+train — Complete

Your model crop_embryo.h5 is successfully trained.

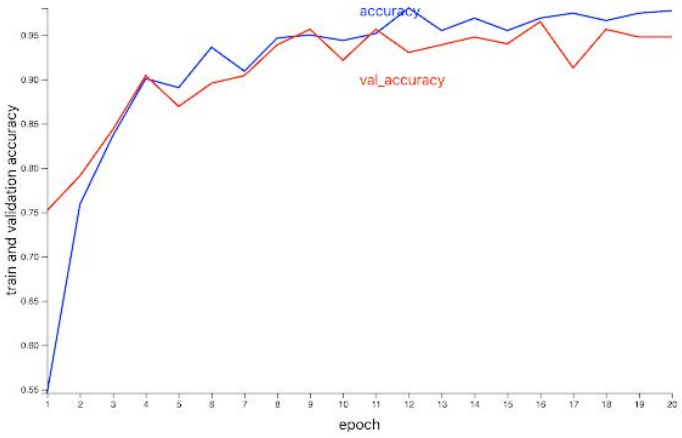


Test accuracy = 0.956
(Click on a cell in the matrix to view image(s) that are predicted differently from their label, or view test data as experiment)

If you are not satisfied with this model, you can restart training or view all trained models of this project

Model details:

- selected base model: VGG16
- training data size: 500
- epoch: 20



possible ways to improve your model:

Add more data
Your trainning data size is 500, which is not a significant amount. Try increasing your sample by providing new data, which could translate into new cases or new features.
Read more about data tactics

Clean Your Data
Can you improve the signal in your data? Perhaps there are missing or corrupt observations that can be fixed or removed, or outlier values outside of reasonable ranges that can be fixed or removed in order to lift the quality of your data.
Read more about data quality, scope and quantity

Select features and examples
If estimate variance is high and your algorithm is relying on many features, you need to prune some features for better results by picking those with the highest predictive value is advisable.

Hyperparameter Tuning
Finding the best hyper-parameters is usually done manually. It's a simple task of trial and error, with some intelligent guesstimating. You'll simply try as many hyperparameter settings as you have time for, and see which one gives you the best results.
If you model's learning curve does not plateau, try add more epochs to your next training.
Read more about hyperparameters
You may also experiment with the size of your batches and the number of training epochs. Read more about Hyperparameter Optimization

Discard model Rename model Mark as favorite