PowerAI Vision demo with Chocolates

# Accessing PowerAI Vision

It is easy (and free) to get a running instance of AI Vision on the SuperVessel service. You can sign up and start using this immediately at: <https://ny1.ptopenlab.com/AIVision/index.html>

There are some limitations to this service, so it is advised to use it only for simple demos and test workloads, and not for any production use cases. For instance, there are limits on the amount of data that you can work with, and many of the APIs generated will be time limited.

# Building a classification model

In advance, create a new data set for classification. In that data set, create the following categories:

* Caramel
* Crème Egg
* Dairy Milk
* Éclair
* Fudge
* Twirl

Upload the contents of each of the relevant folders for each category. There are around 20 images of each chocolate brand, except Caramel, where there are 50.

During the event, show the data set and start a training model for classification using this data set. Use the time this will take (~5 minutes) to explain what is going on under the covers:

* The images are resized to ensure consistency.
* The images are split into train and test data sets.
* The model is primed with random weights and biases.
* The model is trained using the train subset of images multiple times.
* At regular intervals, the result is tested against the test data set, and corrections are made in the next training iterations.
* After the total number of iterations, we get a model with the lowest loss, and highest accuracy.

Then you can show how simple it is to convert that trained model into a webAPI, and start testing images to show the result. There are test images for Caramel, Crème Egg, and Fudge that are not in the training data set that you uploaded. After that, test with the Wispa test image - which should not be recognised correctly. Explain that this is because we have not trained the model with any images of Wispa chocolates, and in fact there is no category for Wispa. The system cannot recognise something it has not been shown.

At this point, you can demonstrate how easy it is to add an extra category (“Wispa”), and upload the pictures from that folder. You can then start training again, to build a better model. This demonstrates how easy it is to refine and retrain your model to provide extra capabilities.

# Building an object detection model

In advance, upload the zip file ChocolateDetectionModel.zip as a data set. This provides 150 images of chocolates that have tagged bounding boxes for the different brands of chocolate. This saves you the time and effort of tagging the images yourself. I would also recommend that you run the training in advance, as it can take over an hour to train this model using a SuperVessel instance.

You can show the data set of images to the audience, and demonstrate how easy it is to add or remove tags from individual images. You may need to zoom out of the pictures to make it easier to see what is happening.

The next step is to show that it is just as easy to start training with this type of data set to build an object detection model. Once you have set the training running, I would recommend moving to the pre-built model, and use it to create a new webAPI. This can be tested with the test images provided, both for individual chocolates and for multiple chocolates in an image.

# Showing object detection in video

You can use the object detection model you just created to find chocolates in video files as well. Import the provided test video files into the Video Data Platform. You can then apply the model to the video files. I would recommend processing this in advance, to avoid waiting during the event. You will need to have deployed the object detection model as a webAPI before using it on video.