

## PowerAI Vision / Visual Insights Workshop

### Lab exercise 1 - Image classification

In this exercise you will practice Image Classification. With Image Classification, a ML algorithm learns to assign categories (classes) to images. The goal of the exercise is to familiarize with the user interface, Image Classification related tasks and the overall process. We will use a subset of the [Stanford Dogs](#) dataset.

The datasets used in the lab are available on github (the instructor shares the link) – download the files to your computer: boxer.zip, chihuahua.zip., cvb\_test.zip

1. **Sign On** to PowerAI Vision.
2. Create a **data set** called CvB (for Chihuahuas vs Boxers)
  - a. On the welcome screen, press 'Get Started' (alternatively, select Data Sets from the menu bar). The Data Sets screen appears.
  - b. Press the plus sign on the leftmost tile to 'Create a new dataset'. Name the data set 'CvB'.
  - c. Click on the tile 'CvB'. A 'Data set / CvB' screen will open.
  - d. Click on 'Import files' under the tile 'Drop files here'. (Use 'Import' for opening the file browser; you can select .zipped collections, or separately browsed images)
  - e. Select the file boxer.zip and upload it. The status message will inform you about the success.
  - f. At the bottom of the pane, select 'Items per page: 100' to view as many images as possible. This will improve productivity.
  - g. Hint: If you do not see what you expect, wait some seconds, then press Refresh in the (tertiary) menu bar right above the images (do not refresh the browser window!).
3. Create and assign **categories (classes)**
  - a. Select all pictures by clicking the checkbox in the (tertiary) menu bar right above the images.
  - b. Select 'Assign category' and specify 'Boxer' as the class. A new category will be created and selected images will be labeled with the category name.

- c. On the leftmost pane, Click on Categories and select 'Uncategorized'. The right pane will become empty.
- d. Upload the chihuahua.zip file using the previous method.
- e. After successfully uploading them, select all images (do not forget to specify 'uncategorized' – unlabeled images will have no category assigned, and this helps you easily assign class names with as few clicks as possible. Any other manual method will likely mean much more work.)
- f. Unselect some (4) of the images, and assign the new category 'Chihuahua' to the selected ones.
- g. Assign the remaining pictures to this category manually one by one – which method is faster?

#### 4. **Augment data**

- a. Select the 'chihuahua' images.
- b. Select All.
- c. Click 'Augment Data'.
- d. Select Horizontal Flip with the parameters unchanged. Select Noise with parameter setting 10, Rotation with 20° degrees and Cropping with 80%.
- e. Observe at the bottom of the window the new dataset size. For each selected image, flipping creates one additional picture, other filters create 5 additional images by applying the filter with 5 different random settings. The filters are not cumulative. If you want cumulative application of filters, you need to process the images in separate steps (create one augmented dataset with the first filter, then another one with the second etc.).
- f. Click Continue.
- g. Name the new dataset 'CvB\_aug'.
- h. Once augmentation is complete, a new data set is created. This new dataset contains the original images **plus** the augmented ones. Successful augmentation preserves the image set information and increases the data set size, at the same time introduces enough variability to decrease high variance errors.
  - i. Note: Augmentation is one part art, one part science. Experiment with the filters to see which ones help you most in the given situation.

Using filters without control will likely deteriorate the model instead of improving it.

**5. Train your model**

- a. You will train two models one based on the raw data set, another one based on the augmented data set. First select the dataset 'CvB'.
- b. In your dataset, press the 'Train' button.
- c. Name the new model 'CvB\_classifier'.
- d. Select Image Classification as the Type of Training.
- e. Set 'Advanced settings' to 'On' in the top-right corner of the screen.
- f. For the 'CvB' model, select 'Training options/Enable Core ML' in 'Optimize model using/System Default (GoogLeNet)'. This will generate the assets needed for CoreML inferencing. We will not use these assets, but get acquainted with the process to generate them.
- g. Set 'max iteration' to a value between 600 and 800.
- h. Leave other parameters unchanged.
  - i. Note: Advanced parameters can improve but can also prevent successful training. Base models are pretrained models that can speed up the training process if there are similarities between the actual data set and the base model.
- i. Press 'Train' to schedule training, and monitor the training process. After initialization the training starts and training loss is shown vs iteration. Training loss and accuracy improves, whereas Test Loss varies.
- j. Once Train Loss is eliminated, your model learned what can be learned from this data. Test Loss may not be very good. This can be a sign of overfitting (insufficient data). Wait until the model training completes.
- k. Repeat steps b.-h. for the other dataset 'CvB\_aug'.

**6. Evaluate the model.**

- a. Repeat for both models (find them in 'Models' menu that is located in the primary menu bar):
- b. Click 'Model details'
- c. Check the training results (evaluation metrics) and the confusion matrix (it is only visible if the Advanced metric option is set to 'On').

- d. Observe the Categories and the included images. You may see if one category is overrepresented (for example due to augmentation). In such cases there is a chance that the classifier becomes skewed. (Understands the category much better than the rest). It would be a good idea to equally augment each category.
  - i. Note: Exported models can be saved or transferred to another PAIV machine (e.g. for inference). The model is created as a password-protected .zip file prior to PAIV1.1.5. The password is undisclosed, and the .zip file cannot be used elsewhere. Beginning with PAIV 1.1.5, the password is not used any longer, but the exported model can be used only in PAIV / VI.
  - ii. Note: Custom models based on Tensorflow can also be exported. They can be deployed anywhere, not only on PAIV systems.

#### 7. **Export** the model

- a. Under Models, select the one you just created and click 'Export model'. It will be saved on your machine.

#### 8. **Deploy** and **test** your classifier.

- a. Click 'Deploy model'. Assign a name to the deployed model.
- b. After the deployment finishes, click the name of your newly created model.
- c. Observe that an API endpoint is created that accepts your service requests and is ready to use the classifier for categorizing the data sent.
- d. Upload an image from the input data.
- e. The model assigns a category, a confidence level and shows on the heat map which part of the image contributed most to the final decision.
  - i. Note: this training, although accuracy could have reached a high value, may not be an excellent predictor. It produces errors and does not generalise well.
- f. Test with different images.

#### 9. **Import** the model

- a. From the Deployed models menu option, select the deployed model and undeploy it. Under models, select the model and delete it (Only delete the one you have exported in the previous step).

- b. Under Models, select Import zip file, and specify the previously saved model. Deploy and test it. It should behave exactly the same as previously.
- c. Try to exchange models with other class members.

#### 10. **Exporting and importing** the datasets.

- a. From the labeled dataset (click on the tile under Data Sets) select export and save the exported dataset on your desktop.
- b. Extract the zipped file – do not delete the zipped version.
- c. Examine the zip content, the filenames and peek into the 'prop.json' file.
- d. Delete the dataset on PAIV.
- e. Import the labelled dataset from the .zipped file. Check that all labels are restored.
  - i. Note: Importing external labelled datasets has some limitations. Refer to the documentation page at:  
[https://www.ibm.com/support/knowledgecenter/SSRU69\\_1.1.5/base/vision\\_dataset\\_consider.html](https://www.ibm.com/support/knowledgecenter/SSRU69_1.1.5/base/vision_dataset_consider.html)

#### 11. Create a **project**.

- a. A project is a collection of related models and data sets that can be used for efficient workflows.
- b. On the **Projects** page, click the large 'plus' button, 'Create new project'
- c. Assign a name: 'CvB'.
- d. Click on the name – the Project / CvB page is displayed.
- e. Add datasets to the project ('CvB').
- f. Add models to the project ('CvB').
- g. Go to 'Models', and set the 'CvB' model status to 'Production' (Select it and then mark it as 'Production ready').
  - i. Note: projects can be used in a production workflow to automatically deploy successfully tested models (that are the 'latest, best'), and keep unsuccessful, rejected models in the improvement cycle. Setting production workflow and automatic deployment requires CL or API calls, and not supported on the GUI. Refer to the documentation for further details.

This concludes the exercise.