

## PowerAI Vision Workshop Exercise 2 – Object detection

In this exercise you will practice Object Detection. With Object Detection, a ML algorithm learns to locate and identify defined objects in a scene. It is a sophisticated and very useful technique for several AI applications. The goal of the exercise is to familiarize with the user interface, the pipeline and OD related tasks.

1. Create a **dataset** called Buildings.
  - 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 to 'Create a new dataset' on the leftmost tile. (.zip file upload is valid only for datasets exported from PAIV). Name the data set 'Buildings'.
  - c. Click on the tile 'Buildings'. A 'Data set / 'Buildings' screen will open.
  - d. Click on 'Import files' under the leftmost tile 'Drop files here'. Select all but a couple of the input images.
  - e. The 'Buildings' data set is created. Duplicate the Data set with a new name 'Building\_g\_b' (for good and best).
  - f. At the bottom of the pane, select 'Items per page: 100' to view as many images as possible. This will likely improve productivity.
2. Labeling **preparation**. Because labeling objects is tedious, we would use an iteration: with minimal labeling, train a simple, rough model, then we will use it to help label more images and do manually only the correction.
  - a. Open the 'Building\_g\_b' dataset.
  - b. Select all the images that do not contain towers on the buildings. Delete them.
  - c. Create two categories: 'good' and 'best'. We will use these to organize images.
  - d. Select 10-15 images that are the most relevant and have clear visibility for towers on the building. Assign the category 'best' to these images.
  - e. Select another bunch (~20) images that are fairly well representing the towers. Assign the category 'good'.

- f. From categories, unselect 'good' and 'best'. Only uncategorized images are shown. Select all and delete them.
  - g. The dataset now contains only the good and the best images. We will use it in phase 2 of the iterative training.
  - h. Go back to the Data Sets screen.
  - i. Duplicate the data set with a new name 'Buildings\_b' (for best).
  - j. Select the category 'good' images and delete all.
- 3. Define and **label** objects.
  - a. On each image, perform the following steps.
    - i. Click on the image and select 'label objects'
    - ii. If there is no object category, select 'Add new' and create an object. You will use 'tower' and 'chimney' objects.
    - iii. Select the object tag and draw a bounding box around the object.
    - iv. All objects of which the majority is visible should be labeled. Do not label objects that are not clearly representing that you want to recognize. Do not leave much empty space around the object. Do not let the bounding box get out of the image boundary.
  - b. Once you finished labeling, augment the data with horizontal flipping and color. Create a new dataset e.g. 'Building\_b\_aug'.
- 4. Select the new dataset and start **training** for Object detection (R-CNN)
  - a. *Note: For object detection, the training can be stopped without losing the model. Once training loss flattens, there is no need to wait more, because the model will not improve.*
- 5. Once the model is ready (it will not be a very good model, we just hope it is better than manual work), deploy it.
- 6. Go back to Data sets and work with 'Buildings\_b\_g'.
  - a. Select Auto Label, and from the deployed models list specify the recently trained and deployed model.
  - b. Let Autolabel run for appx. 30 seconds.
  - c. On each image, perform the following steps:
    - i. Click on the image and select 'label objects'

- ii. If there is no object category, select 'Add new' and create an object. You will use 'tower' and 'chimney' objects.
  - iii. Check and correct the object bounding boxes and labels assigned by auto labeling.
  - iv. Complete labeling for missing boxes and objects.
  - v. Select the object tag and draw a bounding box around the object.
  - vi. All objects of which the majority is visible should be labeled. Do not label objects that are not clearly representing that you want to recognize. Do not leave much empty space around the object. Do not let the bounding box get out of the image boundary.
- d. Once you finished labeling, augment the data with horizontal flipping and color. Create a new dataset, e.g. 'Building\_b\_g\_aug'.
- 7. Select this dataset and start **training** for Object detection (R-CNN). Before that, if you need to free up resources (training will require a GPU), you can delete the previously deployed model (it was bad anyway).
  - a. *Note: For object detection, the training can be stopped without losing the model. Once training loss flattens, there is no need to wait more, because the model will not improve.*
- 8. Once the model is ready (it will still not be a very good model), deploy it.
- 9. Go back to Data sets and work with 'Buildings'.
  - a. Select Auto Label, and from the deployed models list specify the recently trained and deployed model.
  - b. Let Autolabel run for appx. 30 seconds.
  - c. On each image, perform the following steps:
    - i. Click on the image and select 'label objects'
    - ii. If there is no object category, select 'Add new' and create an object. You will use 'tower' and 'chimney' objects.
    - iii. Check and correct the object bounding boxes and labels assigned by auto labeling.
    - iv. Complete labeling for missing boxes and objects.
    - v. Select the object tag and draw a bounding box around the object. All objects of which the majority is visible should be labeled. Do not label

objects that are not clearly representing that you want to recognize.

Do not leave much empty space around the object. Do not let the bounding box get out of the image boundary.

- d. Once you finished labeling, augment the data with horizontal flipping and color. Create a new dataset, e.g. 'Building\_aug'.

10. Delete the deployed model if you need to free up a GPU for training.

11. Train the model using the 'Building\_aug' dataset. Wait for the training loss to stabilize below your acceptance level.

12. Deploy the model. It should be a fair model.

13. Test the model for object detection using images that were not involved in the training process.

This concludes the exercise.