Code Sample - Find Phone - Machine Learning & Vision

SSD: Single Shot MultiBox Detector

- Usage:

  As advised by the task description, there are 8 python scripts in total. Please put those scripts under the same folder. Those two scripts advised by the tasks are:
  1) *train_phone_finder.py* takes a single command line argument which is a path to a folder with labeled images and *labels.txt* that has been attached to this description. This script trains the model and generates a checkpoint file *checkpoint_ssd300.pth.tar*, which contains all the parameters of the model for predicting testing data. Here is what a terminal command would look like:

  ```
  > python train_phone_finder.py ./find_phone
  ```

  2) *find_phone.py* takes a single command line argument which is a path to the jpeg image to be tested. This script uses *checkpoint_ssd300.pth.tar* previously generated by train_phone_finder.py, so please make sure the checkpoint file and this script are in the same folder. Here is what a terminal command will look like:

  ```
  > python train_phone_finder.py ./find_phone/39.jpg

  0.61358 0.33051
  ```

Description:

For this task, I reference to the Single Shot MultiBox Detector (SSD) proposed by Wei Liu et al.[1] SSD utilizes a single deep neural network and has high accuracy on multi-class objection detection tasks. So I believe it can also have great performance on single-class objection detection.
Most of the structure follows the open source GitHub repository “SSD: Single Shot MultiBox Detector | a PyTorch Tutorial to Object Detection”[2] . However, the object labels of the dataset used by the paper are presented in bounding boxes. The labels of our phone dataset only has the location of the object center. Therefore, the simplest way for adding a bounding box to each phone in the dataset for training the SSD is to use a fixed-size bounding box for every image. Here, I add the bounding box labels to the phone dataset and train the model from scratch on the phone dataset. The rest of

[1] arXiv:1512.02325 [cs.CV] https://arxiv.org/abs/1512.02325
[2] https://github.com/sgrvinod/a-PyTorch-Tutorial-to-Object-Detection#evaluation

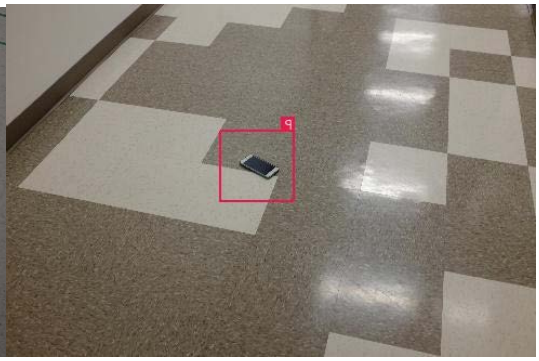
the model follows the open source codes. For detail structure and pipeline of the model, please refer to the original paper and/or the GitHub repository.

Evaluation:

For evaluating the performance of the model, I randomly pick four images as testing set, and the rest of them are training set. In total, there are 125 images for training and 4 images for testing. Below are the parameters and environment used:
Iterations: 1000
Batch size: 8
Learning rate: 0.001
Momentum: 0.9.
GPU: NVIDIA GTX 1080 Ti
CUDA verison: 10.2
CuDNN verison: 7.6.5

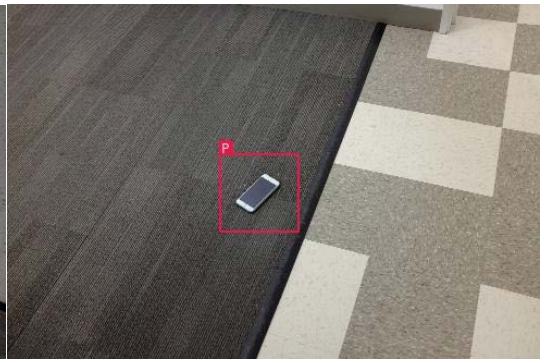
It took less than 7 minutes to train the model on training set and the results are beyond my expectation. We can see that every phone in testing image is accurately marked inside a bounding box! With these boxes, it is easy to transform the information back to the center location.

39.jpg



46.jpg

47.jpg



123.jpg