Project 03 (Final Course Project)

- You can propose your own projects and work on it once agreed by both of you and me. And, make sure you can finish it.
  - E.g., take a look at what have been done in Stanford CS231 class, and propose your own creative projects.
- Requirement: final write-up (15%) and self-contained reproducible code (25%)
- Bonus points for oral project presentation (on 5/1/2018, 1-4 pm, in class): 5
- You can work on it individually or join a small group (no more than 3 members and individual contributions need to be clearly stated in the write-up. If more than 3, you need to justify it a bit more).

Project 03 (Final Course Project)

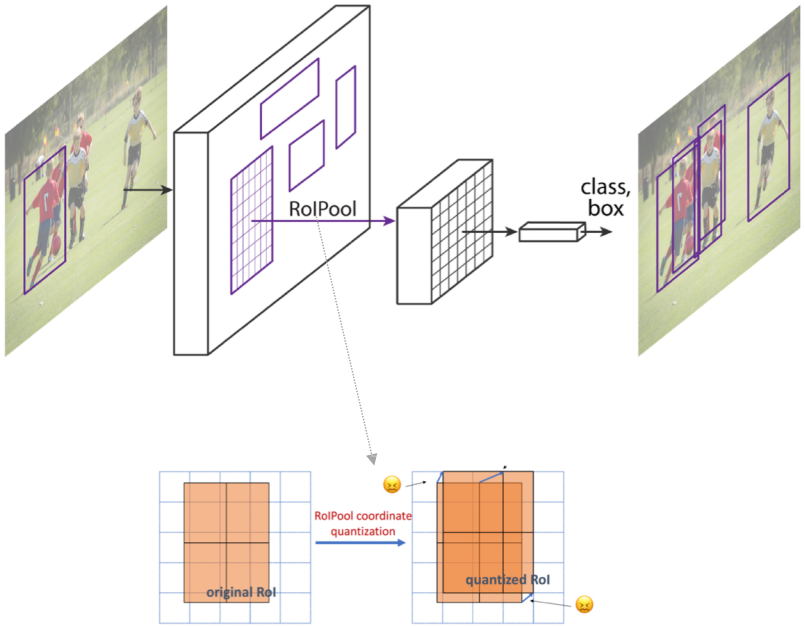
- High-end GPU(s) required:
  - Topic 1: Object detection based on Faster-RCNN (Recommended)
  - Alternative 1: Semantic Segmentation
  - Alternative 2: Visual Object Tracking

  - Topic 2: (your own proposals)

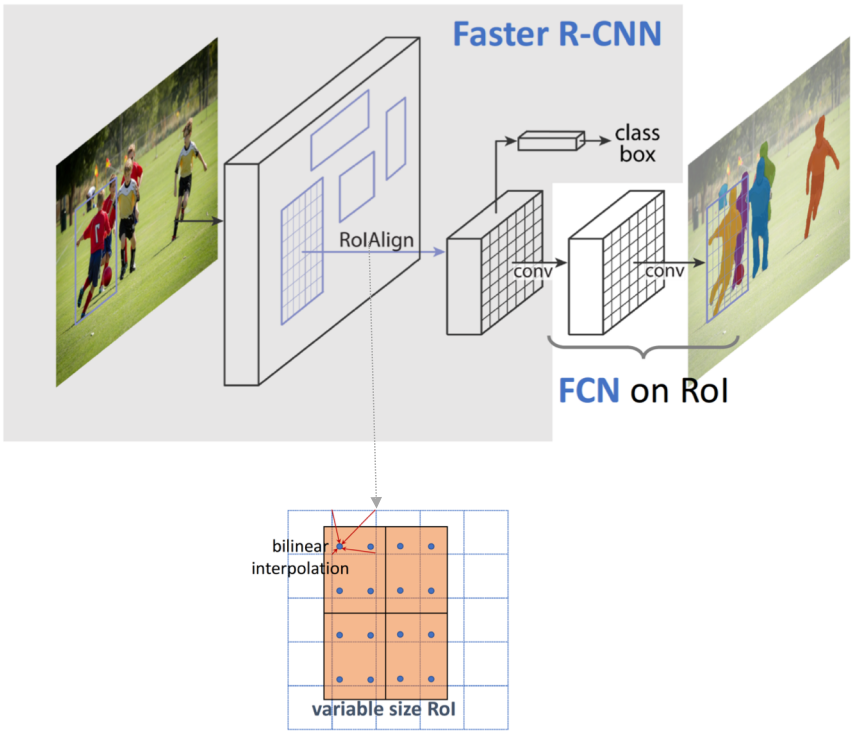
Project 03 (Final Course Project)

- Topic 1: Object detection based on Faster-RCNN

Fast/Faster R-CNN



Mask R-CNN = Faster R-CNN + FCN on ROIs



Ross Girshick. “Fast R-CNN”. ICCV 2015.
Shaoqing Ren, Kaiming He, Ross Girshick, & Jian Sun. “Faster R-CNN: Towards Real-Time Object Detection with Region Proposal Networks”. NIPS 2015.
Kaiming He, Georgia Gkioxari, Piotr Dollár, and Ross Girshick, “Mask R-CNN”, ICCV 2017

Project 03 (Final Course Project)

- Topic 1: Object detection based on Faster-RCNN
  - Existing codes:
    - Pytorch: https://github.com/jwyang/faster-rcnn.pytorch
    - MXNet: https://github.com/apache/incubator-mxnet/tree/master/example/rcnn
    - Tensorflow: https://github.com/endernewton/tf-faster-rcnn
    - MatConvNet: https://github.com/albanie/mcnFasterRCNN
    - Or any others

Important: Read this blog which explains details of faster R-CNN with visualization http://www.telesens.co/2018/03/11/object-detection-and-classification-using-r-cnns/

Project 03 (Final Course Project)

- Topic 1: Object detection based on Faster-RCNN
  - Task 1 (40%): Getting familiar with the existing code which you pick by training it on the face detection tasks.
    - All you actually need to do: prepare your data in the PASCAL VOC format so you can reuse the existing data i/o code.
    - But it will better to really look into the code and understanding it so you can do Task 2.

Project 03 (Final Course Project)

- Topic 1: Object detection based on Faster-RCNN
  - Task 2 (60%): Reproduce (and/or extend) this paper https://arxiv.org/abs/1803.03243 .

Domain Adaptive Faster R-CNN for Object Detection in the Wild

Yuhua Chen[1] Wen Li[1] Christos Sakaridis[1] Dengxin Dai[1] Luc Van Gool[1,2]
[1]Computer Vision Lab, ETH Zurich [2]VISICS, ESAT/PSI, KU Leuven
{yuhua.chen,liwen,csakarid,dai,vangool}@vision.ee.ethz.ch

Abstract

Object detection typically assumes that training and test data are drawn from an identical distribution, which, however, does not always hold in practice. Such a distribution mismatch will lead to a significant performance drop. In this work, we aim to improve the cross-domain robustness of object detection. We tackle the domain shift on two levels: 1) the image-level shift, such as image style, illumination, etc., *and 2) the instance-level shift, such as object appearance, size,* etc. *We build our approach based on the recent state-of-the-art Faster R-CNN model, and design two domain adaptation components, on image level and instance level, to reduce the domain discrepancy. The two domain adaptation components are based on $\mathcal{H}$-divergence theory, and are implemented by learning a domain classifier in adversarial training manner. The domain classifiers on dif-*

Figure 1. **Illustration of different datasets for autonomous driving:** From top to bottom-right, example images are taken from: *KITTI*[17], *Cityscapes*[5], *Foggy Cityscapes*[49], *SIM10K*[30]. Though all datasets cover urban scenes, images in those dataset vary in style, resolution, illumination, object size, *etc*. The visual difference between those datasets presents a challenge for applying an object detection model learned from one domain to another domain.

Project 03 (Final Course Project)

- Topic 1: Object detection based on Faster-RCNN
  - Task 2 (60%): Reproduce (and/or extend) this paper https://arxiv.org/abs/1803.03243 .
    - Why this?
      - This hands-on experience will be of most important when you need to transfer existing deep learning methods to solve your own tasks.
      - This is the state-of-the-art of domain-adaptation method in object detection.

Project 03 (Final Course Project)

- Alternative 1: Semantic Segmentation
  - Paper: https://arxiv.org/abs/1803.06815
  - Ref Code: https://github.com/sacmehta/ESPNet (pytorch)
  - Requirement:
    - 1) it is ok to reuse the pytorch code, but you need to identify a component in the model which would like to modify or try different options. You can not just run the code to get results.
    - Or 2) you can rewrite the code in another platform, e.g., Tensorflow or MXNet, and you need to match the performance.

Project 03 (Final Course Project)

- Alternative 2: Visual Object Tracking
  - Paper: http://openaccess.thecvf.com/content_cvpr_2017/html/Valmadre_End-To-End_Representation_Learning_CVPR_2017_paper.html
  - Ref Code: https://github.com/bertinetto/cfnet (Matlab)
  - Requirement:
    - 1) it is ok to reuse the Matlab code (computer vision package is required), but you need to identify a component in the model which would like to modify or try different options. You can not just run the code to get results.
    - Or 2) you can rewrite the code in another platform, e.g., Pytorch, Tensorflow or MXNet, and you need to match the performance.

Project 03 (Final Course Project)

- Topic 2: (you pick one which interests you mostly)
  - Paper: Find some reference paper to reproduce the results
  - Ref Code: Find the reference which have ref code available.
  - Requirement:
    - 0) The topic needs to include computer vision components (plz discuss with me via email or slack or in person).
    - 1) it is ok to reuse the provided code, but you need to identify a component in the model which would like to modify or try different options. You can not just run the code to get results.
    - Or 2) you can rewrite the code in another platform, e.g., Pytorch, Tensorflow or MXNet, and you need to match the performance.