Intro to Image Understanding (CSC420)
Projects

Report Submission Deadline : April 15, 11.59pm, 2021

Max points for project: 40

Projects can be done individually or in pairs. If a project is done in a pair, each student will still need to defend their part. From the report it should be clear what each student contributed to the project. By April 15 you will need to hand in the project report including code. Make the code organized and documented, possibly including scripts that run your pipeline. In the oral defense you'll need to run some of your code and be able to defend it, as well as answer questions about class material. The oral defense is scheduled in the week of April 19. The grade will evaluate a project report and an oral presentation, at the total of 40% of the final grade.

Whenever you use ideas, code or data from a paper, forum, webpage, etc, you need to cite it in your report. Whenever you use code available from some paper or method, you need to include a short description of the method showing your understanding of the technique. If there were multiple options for techniques or code, please also explain why you chose a particular one.

The grade will take into account the following factors:

- Your ideas to tackle a problem: how appropriate the techniques you chose are for the problem. Coming up with novel ideas is obviously a big plus.
- Your implementation: the accuracy of the solution, speed, and partly also how organized the code is (scripts that run the full pipeline or specific subtasks, documentation). The grade will take into account also the performance of your solution with respect to other students' results.
- Whether you implemented a technique yourself or used code available online
- How you present your results in the report: ideally you would show your results for each task by including a few pictures in your report. Even more ideally, you would show good cases where your method works and also bad cases where it doesn't. Providing some intuitive explanation of the failure cases is a plus.
- Thoroughness of your report: How well you describe the problem and the techniques you chose, how well you analyzed your results and whether your citations to the techniques you used are appropriate.

Do **not** put the video clips we provide online or share with anyone.

Project 1

This project is about analysis of news broadcast. You will be given a news video clip. Here are the tasks to solve:

(a) Detect shots in the videos. A shot is a set of consecutive frames with a smooth camera motion.

(b) (Manually) Annotate shot boundaries in the video. How would you evaluate how well you are detecting the shots? Why is this a good metric? Compute the performance according to your proposed evaluation metric.

(c) Detect the news company's logo. Please do not assume that you know the location of the logo within a frame, it should be found automatically.

(f) Detect faces in the video.

(g) Perform face tracking by correctly associating a face detection in the previous frame to a face detection in the current frame.

(i) You will be given a dataset of female and male faces. Train a classifier that can predict whether a face is female or male. For each face track in the news video predict whether it is female or male. To do this you will take a few images of faces, compute image features and train a male-vs-female classifier, e.g., SVM, NN. Once trained, you will predict the gender of each face detection in the video. That is, you'll take each face detection (a crop in the image specified by the face box), compute appropriate image features, and use your classifier to predict the gender of the face. How would you decide whether a full face track is female or male?

(j) Visualize your results: produce a video in which you show a bounding box around the detected company logo, and bounding boxes around the detected faces. Each face bounding box should have text indicated whether the face is male or female.

Useful Code and Techniques

Shot Detection. A simple way of detecting shot boundaries is to look at differences in color histograms between two consecutive frames. An even better way is to look at Displaced Frame Distances, described in Section 2.1 of this paper:

Makarand Tapaswi, Martin Baeuml and Rainer Stiefelhagen, *"Knock! Knock! Who is it" Probabilistic Person Identification in TV Series*, CVPR 2012, `https://cvhci.anthropomatik.kit.edu/~mtapaswi/papers/CVPR2012.pdf`

More options are discussed in this paper:
Y. Yusoff, W. Christmas, and J. Kittler, *A Study on Automatic Shot Change Detection. Multimedia Applications and Services*, 1998, `http://www.cs.utoronto.ca/~fidler/slides/CSC420/papers/shots.pdf`.

Face Detection. A few options:

- The first paper that did great on faces was Viola-Jones face detector:
  Paul Viola and Michael Jones, Rapid object detection using a boosted cascade of simple features, CVPR, 2001, `https://www.cs.cmu.edu/~efros/courses/LBMV07/Papers/viola-cvpr-01.pdf` There is lots of code online.

- Paper:
  P. Felzenszwalb, R. Girshick, D. McAllester, D. Ramanan, *Object Detection with Discriminatively Trained Part Based Models* IEEE Transactions on Pattern Analysis and Machine Intelligence, Vol. 32, No. 9, Sep. 2010 `http://cs.brown.edu/~pff/papers/lsvm-pami.pdf`
  Code: `http://www.rossgirshick.info/latent/`
  with a trained model called `dpm_baseline.mat` available here:
  `https://bitbucket.org/rodrigob/doppia/src/tip/data/trained_models/face_detection/?at=preparing_v2`. This detector may work a little better. You can also check a detailed analysis and code for different face detectors here: `http://markusmathias.bitbucket.org/2014_eccv_face_detection/`

- Paper:
  X. Zhu, D. Ramanan. *Face detection, pose estimation and landmark localization in the wild*, CVPR 2012, `http://www.ics.uci.edu/~xzhu/face/`.
  This detector however also gives you the keypoints for the facial landmarks.

Tracking Here are some options (papers), choose the one it suits you best:

- Tracking via dynamic programming: `https://engineering.purdue.edu/~qobi/papers/acs2012b.pdf`

- Tracking via Hungarian method: `http://www.cvlibs.net/publications/Geiger2014PAMI.pdf`, first 3 paragraphs of Section 4.1

- Tracking via Kalman filter: lots of tutorials online

Available code:

- Check http://www.ics.uci.edu/~dramanan/ for paper (and code): 'H. Pirsiavash, D. Ramanan, C. Fowlkes. "Globally-Optimal Greedy Algorithms for Tracking a Variable Number of Objects", Computer Vision and Pattern Recognition (CVPR), 2011
- http://www.cvlibs.net/software/trackbydet/
- http://research.milanton.de/dctracking/

Stereo For autonomous driving, it's best to check the stereo challenge of the KITTI dataset (http://www.cvlibs.net/datasets/kitti/). It has links to code and gives you running time of each algorithm. Some options:

- http://userpage.fu-berlin.de/spangenb/
- http://ttic.uchicago.edu/~dmcallester/SPS/index.html

Flow For autonomous driving, it's best to check the stereo challenge of the KITTI dataset (http://www.cvlibs.net/datasets/kitti/). It has links to code and gives you running time of each algorithm.

- http://people.csail.mit.edu/celiu/OpticalFlow/
- Matlab has some functions to compute flow: http://www.mathworks.com/help/vision/ref/opticalflow.html
- OpenCV has flow and trackers, and other useful stuff: http://opencv.org/documentation.html

Other For features, super pixels, classifiers, etc, look under Resources of class webpage: http://www.cs.utoronto.ca/~fidler/teaching/2019/CSC420.html