Object Detection: Lecture 3

Ujjwal
ujjwal.ujjwal@inria.fr

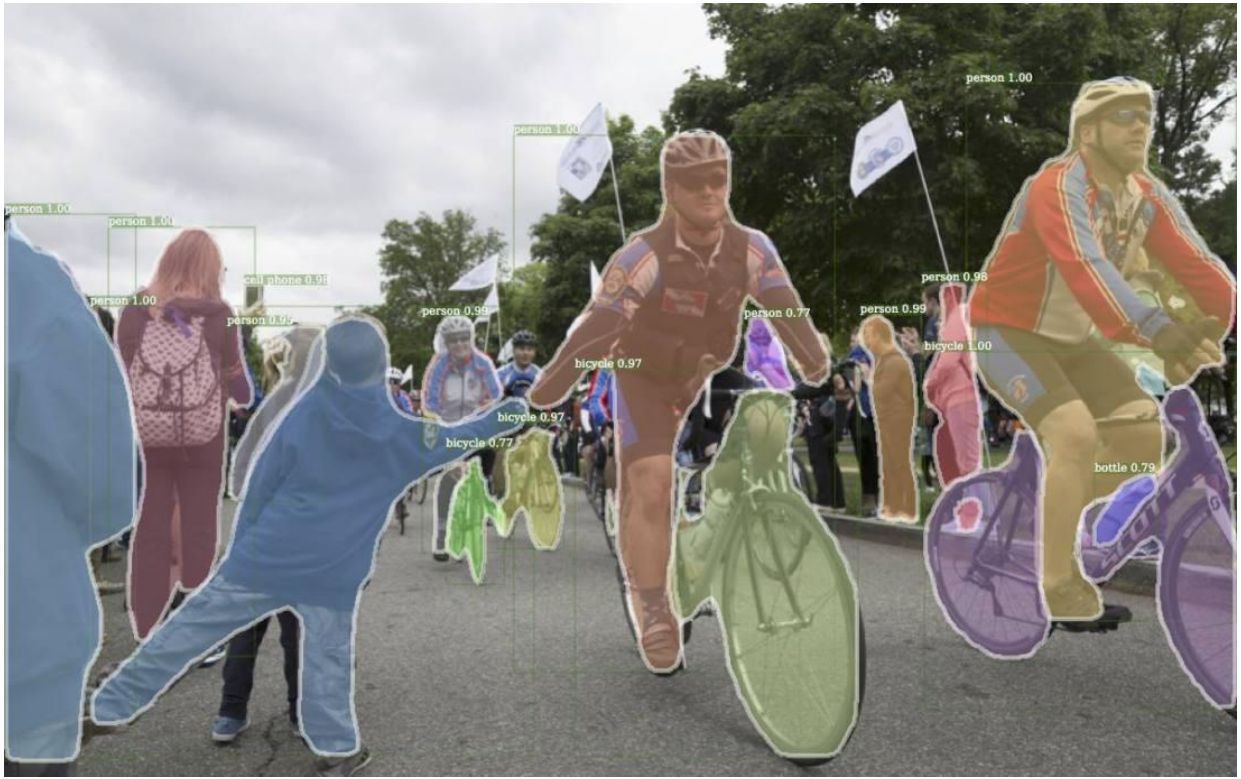
Previous Lectures

- Faster-RCNN
- SSD
- Feature Pyramid Networks

This Lecture

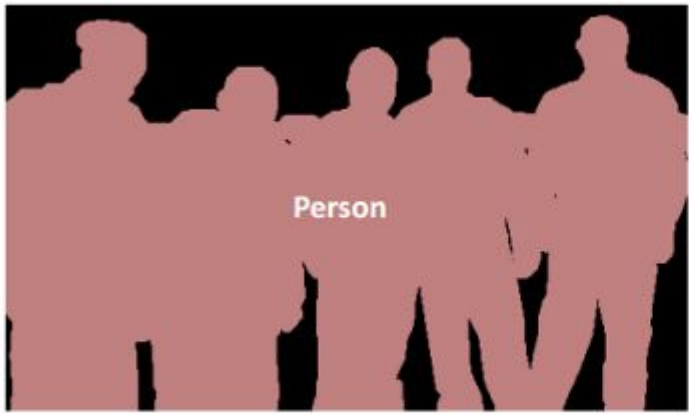
- Mask-RCNN
- Practical View of Object Detection

Mask-RCNN

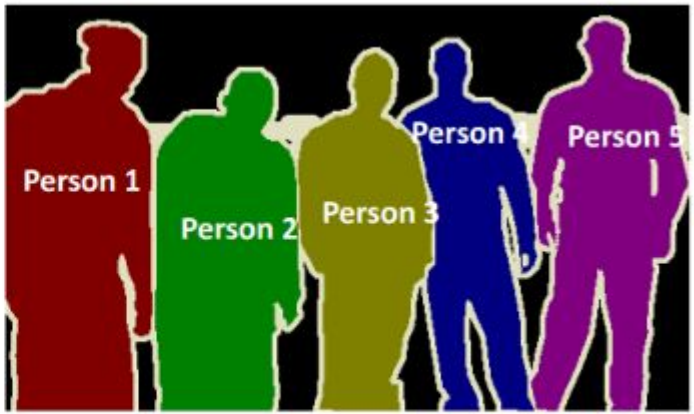


- Object Detection.
- Instance Segmentation.
- KeyPoint Detection.

Instance Segmentation

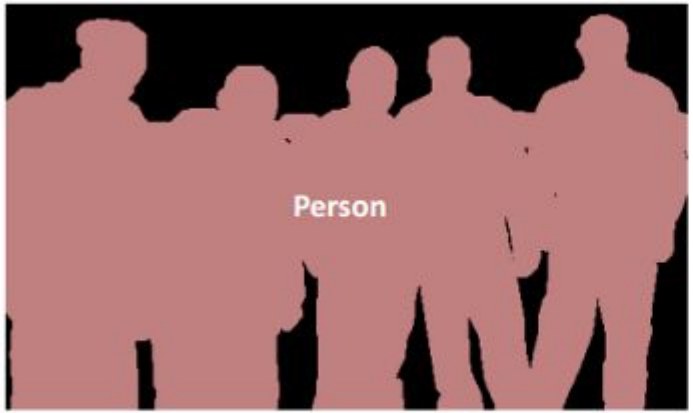


Semantic Segmentation



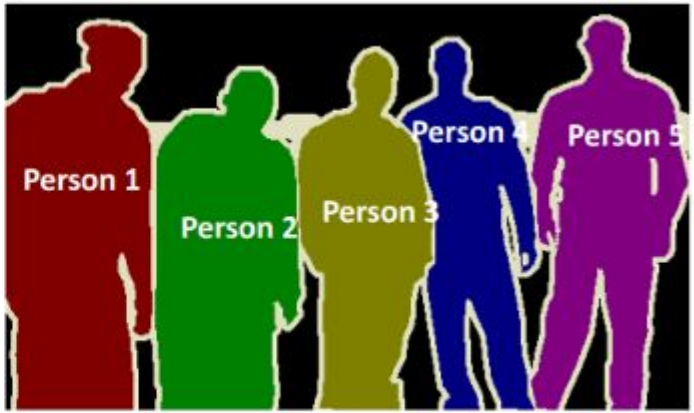
Instance Segmentation

Instance Segmentation



Semantic Segmentation

Class aware but **NOT** instance aware



Instance Segmentation

Class aware **AND** instance aware

Keypoint Detection

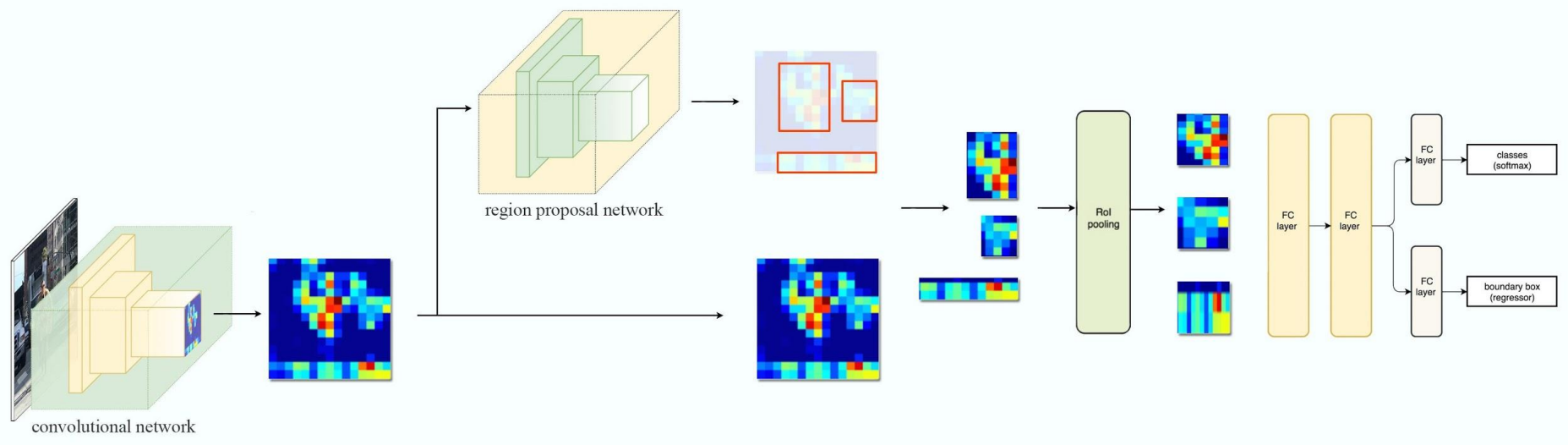
Why Mask-RCNN ?

- Often multiple tasks are desired at the same time.
  - With instance segmentation one can analyze each object individually.
  - With keypoint detection one can analyze poses of people individually.

Then Why not only Mask-RCNN ?

- Overkill is a dangerous habit.
- Practical problems must be solved practically.
    - Sometimes when only bounding box detection is needed why one would go for Mask-RCNN.

Understanding Mask-RCNN



Some Professionally Good Implementations

- Best
  - TensorFlow Object Detection API (For FRCNN, SSD, FPN, Mask-RCNN)
    - Very useful for professional usage.
  - MatterPort Mask-RCNN
    - Very good for Mask-RCNN implementation.
    - Professionally very useful
  -

Which Framework to Use ?

- TensorFlow
  - **Pros**
    - Very well written.
    - Very well maintained.
    - Professionally complete.
  - **Cons**
    - Slightly complex to use and learn.
- PyTorch
  - **Pros**
    - Python like interface.
    - Easy to use.
  - **Cons**
    - Not very consistent.
    - Not well-maintained.

Useful Professional Tips

- Study your data.
  - What classes are there ?
  - How different are they ?
  - What data augmentations make sense ?
- Study the constraints
  - How much memory is available ?
  - Any speed requirements ?
- Experiment Slowly
  - Write a basic implementation first.
  - If using any other implementation, understand it first.
  - Experiment with hyperparameters (e.g: Learning rate, optimizer etc.)

A Basic CoLab Experiment

Beginning Steps

- Open the Git Repo here.
- Clone the repo on your system.
- Open Google CoLab and upload the Ipython notebook in it.
- Upload dog_dataset.zip (in the repo) to your Google Drive.
- Get the FileID of the above zip file in your google drive.
    - To get the file id, get a sharable link to the zip file.
    - The part of the link after “id=” is the file ID.
- In the ipython notebook in the CoLab, under the section “Download and extract dataset”, replace fileID with the ID you received in the last step.

Run the cells

- Here Detection of a dog is being done using TensorFlow Object Detection API.

AMA: Ask Me Anything

- Use the remainder of this class to ask me any sort of question or queries about object detection or deep learning in general.