Exercise for MA-INF 2218
Video Analytics SS21
Submission deadline: 22.06.2021
Action Localization

1 Task

The task is to spatially localize an action that is being performed by a human in a video. Your network should localize the action with a bounding box and classify it into a set of pre-defined action categories. We only focus on spatial localization in this sheet. You have to train your network using the training set, and evaluate it using the testing set. (Details below)

You are going to implement a simple baseline for this task, very similar to [1]. The baseline is a Faster-RCNN detector on a single RGB frame.

The base network should be a ResNet-18 network pretrained on ImageNet. This network is already available with PyTorch. Don't forget the preprocessing necessary for the ResNet which preparing your input to the network.

During training, you are given a single bounding box for each frame and an action category. At test time you have to predict the single action that is being performed. To make it easy, just return the most likely anchor box after bounding box regression.

Please provided a file `detections.json` for each video, similar to `objects.json` (details below) as part of your submission. Please don't submit the video frames.

1.1 Anchor Boxes

Given the dataset you are provided with, please describe shortly in your README file, what would be a good choice for anchor boxes during training (sizes, aspect ratios) for the dataset in question and why. Details on the dataset are provided below.

1.2 Training

In your README, please provide an overview of the training procedure that you used for training your network, as well as values of the hyper-parameters that you used and briefly describe your choice.

2 Dataset

We are going to use the J-HMDB dataset: `http://jhmdb.is.tue.mpg.de/` You don't have to download the dataset, the dataset is already prepared for you in the same folder as this pdf file. The videos in this dataset are trimmed (around 40 frames each) and clipped to the length of the actions. Each video contains a single action performed by a single person.

After extracting the zip file, you can see the *train* and *test* folders. Inside each folder is a set of videos. Inside each video folder there exists a `objects.json` file and a set of RGB frames like `00001.png` and `00002.png`, etc.

The `objects.json` file contains information about the bounding box and action in each frame.

```
{
'frames': [
  {'action': 'catch',          // action name
   'bbox': [129, 53, 39, 74], // [x, y, w, h]
   'file_name': '00001.png',  // file name of the frame this object corresponds to
   'index': 0},               // index of the frame this object corresponds to
  // ...
]
}
```

1. There are 21 action classes. (*Don't forget the background class while designing your network!*)
2. Frame sizes are 320×240 (width, height).
3. Bounding box is specified using $x, y, width, height$ format. The origin is top left corner of the frame.

3 Evaluation

For each frame of a test video, pick the bounding box with the highest score (objectness or actioness score) then apply the bounding box regressions. Given that you already know the ground truth bounding box for that frame in the annotation file, compute the IOU (Intersection over Union) between the predicted bounding box and the ground truth bounding box if the correct class is predicted by the classification head. If the correct class for the action is not predicted assume IOU is zero. Average the IOUs over all of the frames in a video. Then average over all videos and report it in your README. Also, please provide several frames with the visualized predicted and ground-truth bounded boxes.

4 Notes

1. Make sure you write readable and well commented code.
2. Create a README file provide the information you want in there.
3. Please note that you have to implement Faster-RCNN yourself and not allowed to use the network implemented in pytorch. However, you are allowed to use helper modules implemented there: anchor generator, balanced batch sampler, box matcher and the roi-pooling module. If you use any additional code from the web, please specify and cite them in your README file.

5 Good resources to learn about Faster-RCNN detectors in general

1. ICCV 2017 tutorial by Ross Girshick: `https://www.youtube.com/watch?v=TxNDk65R3qI`

2. This blog post: `https://tryolabs.com/blog/2018/01/18/faster-r-cnn-down-the-rabbit-hole-of-modern-object-detection/`

3. And maybe this blog post: `http://telesens.co/2018/03/11/object-detection-and-classification-using-r-cnns/`

4. There are way too many resources on the web, just search.

In case of any question please email the mailing list or reach me (Olga Zatsarynna) directly at **zatsarynna@iai.uni-bonn.de**

[1] Peng, Xiaojiang, and Cordelia Schmid. "Multi-region two-stream R-CNN for action detection." ECCV 2016.