

Fine-grained Recognition with Part-Transfer

Computer Vision Group, Friedrich Schiller University of Jena

Authors: Christoph Goering, Erik Rodner, Alexander Freytag

contact: Erik.Rodner@uni-jena.de

Needed libraries and third party software

1. vlfeat - for extracting features - <http://www.vlfeat.org>
2. yael - fast, multithreaded k-means - <https://gforge.inria.fr/projects/yael/>
3. opencv - only used for grabcut - <http://opencv.org>
4. liblinear - <http://www.csie.ntu.edu.tw/~cjlin/liblinear/>
5. color names - <http://lear.inrialpes.fr/people/vandeweyer/software>

Notes:

- if installed, the parallel toolbox can be used. just uncomment the parfor in vlfeatExtractFeatures and vlfeatCreateCodebook
- mex-wrapper for grabcut needs to be compiled before first use

Usage (standard experiments)

```
recRate = experimentParts('cub200_2011',nrClasses, ...  
    config, configParts)
```

- nrClasses = 200 | 14 | 3
- config - parameters to influence extraction of global features, a list can be found in experimentGeneral_extractGlobalFeatures.m
- configParts - for features extracted from parts, list can be found in experimentGeneral_extractPartFeatures.m

Examples

1. use default values:

```
recRate = experimentParts('cub200_2011',nrClasses, ...  
    struct([]), struct([]))
```

2. do not use global features:

```
recRate = experimentParts('cub200_2011',nrClasses, ...
    struct('useGlobal','no'), struct([]))
```

3. do not use part features:

```
recRate = experimentParts('cub200_2011',nrClasses, ...
    struct([], struct('useParts','none'))
```

4. use part features, estimated using nearest neighbour:

```
recRate = experimentParts('cub200_2011',nrClasses, ...
    struct([], struct('useParts','nn'))
```

5. do not use grabcut segmentation:

```
recRate = experimentParts('cub200_2011',nrClasses, ...
    struct('preprocessing_useMask','none'), ...
    struct('useParts','none'))
```

6. use the k-best part estimations:

```
recRate = experimentParts_knn('cub200_2011',nrClasses, ...
    struct([], struct([]))
```

Details of the algorithm

The algorithm is described in detail in the corresponding paper, here, we just give a very brief overview and mention some additional aspects:

- opoonentSift and colorname features are used
- classification is done using liblinear and an approximated chi square kernel
- global features can be extracted from the whole image, the provided bounding box, or from a grabcut segmentation
- left and right instances of part features are pooled
- for classification all features are concatenated
- part transfer is based on HOG feature matching