Deformable Part Descriptors (DPD)

This code accompanies the ICCV 2013 paper **Deformable Part Descriptors for Fine-grained Recognition and Attribute Prediction**.

Running DPD+DeCAF demos

User Configuration

```
in extract_dpm_parts.sh (or, as an environment variable):
  dpd_scratch=/scratch

in dpd_decaf/prepare_CUB2011_for_ffld.py:
  birdLocation = ./CUB_200_2011 #locate CUB birds dataset
python ./dpd_decaf/prepare_CUB2011_for_ffld.py #convert CUB birds to our data format

TODO: automatically create all necessary directories
```

Running the Code

Here's how to automatically classify CUB200-2011 birds using Deformable Part Descriptors with DeCAF convolutional features:

```
./extract_dpm_parts.sh #calls into ffld_dpm/build/ffld
./dpd_decaf/run_dpd_decaf_features.sh #DeCAF convnet features on DPM parts
matlab
>dpd_decaf; %weak pooling, SVM training, SVM classification

#dpd_decaf is hard-coded for this config: (weak pooling; CUB200_2011; trainAndTest).
# not too hard to modify, though.
```

Running DPD+KDES demos

User Configuration

in `dpd_set_up.m`:

```
scratchdir = /scratch  %for KDES features, DPD features, etc
```

```
if strcmp(database, 'bird')
    dataset_base = /path/to/CUB200-2011 %you edit this
elseif strcmp(database, 'cub200')
    dataset_base = /path/to/CUB200-2010 %you edit this
elseif strcmp(database, 'human')
    dataset_base = /path/to/berkeley-human-attributes-dataset %you edit this
end
```

Running the Code

Here's how to automatically classify CUB200-2011 birds using Deformable Part Descriptors with Kernel Descriptors (KDES):

```
run_dpd('bird', 0, 0); %runs all images through each DPD pipeline step, in batch mode

%arguments: run_dpd('class', 0=weakPooling 1=strongPooling, 0=trainAndTest 1=trainOnly);
%assume: DPM part bounding boxes are already extracted and located in a subdirectory of dpd_scratch
```

Reorganizing directories

```
move:
dpd_set_up -> dpd_kdes_set_up
thirdparty/ffld_dpm ./ffld_dpm

new:
dpd/dpd_decaf
    mv train_test/dpd_decaf.m dpd_decaf/dpd_decaf.m
    mv ./dpd_decaf_features.py dpd_decaf/dpd_decaf_features.py
dpd/dpd_kdes
    mv train_test dpd_kdes
    common/{cksvd*, kdes_cholesky} -> dpd_kdes
dpd/old
    mv dpd/test_realtime old
    mv dpd/test_realtime_cpp old
    mv thirdparty old/thirdparty

    mv fgcomp old/fgcomp  #just a script to convert FGcomp labels to pascal format

add:
dpd_decaf_set_up.sh #just point to dpd_scratch directory location
```