

This is an image classification dataset used for EECS545 final project which belongs to group 10.

1. Source

The data is originally obtained from LSVRC2010 and then modified for appropriate use, as required, we should cite

```
1  @article{ILSVRC15,  
2  Author = {Olga Russakovsky and Jia Deng and Hao Su and Jonathan Krause  
    and Sanjeev Satheesh and Sean Ma and Zhiheng Huang and Andrej Karpathy  
    and Aditya Khosla and Michael Bernstein and Alexander C. Berg and Li  
    Fei-Fei},  
3  Title = {{ImageNet Large Scale Visual Recognition Challenge}},  
4  Year = {2015},  
5  journal  = {International Journal of Computer Vision (IJCV)},  
6  doi = {10.1007/s11263-015-0816-y},  
7  volume={115},  
8  number={3},  
9  pages={211-252}  
10 }
```

2. Overview

The dataset is used for image classification, it contains 100 classes of images, each class is labeled with word net id.

3. Feature Vector

Each image is expressed as a feature vector, which represents the multiplicity of specific "word" in the image, i.e each feature vector is a "bag of words".

Feature vector is of length 1000.

4. Training, Validation and Test

It contains three sets of data, "training, validation and test". "training" contains 500 images of each classes, "validation" contains 50 images of each class, and "test" contains 100 images of each class.

5. How to use

If you want to use the first 20 classes of images in training set

```
1 load('features.mat');
2 % get first 20 classes of images
3 % class_num = 100, train_num = 500, valid_num = 50, test_num = 100
4 train_x_20 = train_x(:, 1: 20 * train_num);
5 train_y_20 = train_y(1: 20 * train_num);
6 % do a permutation before use
7 perm = randperm(20 * train_num);
8 train_x_20 = train_x_20(:, perm);
9 train_y_20 = train_y_20(perm);
```

If you want to know the wordnet_id of a class

```
1 % class is a number between 1 and 100
2 wordnet_id = wordmap(class);
```

6. Normalize before use

According to paper, you might want to normalize features.

"To this end, we use the densely sampled SIFT features clustered into 1k visual words provided by the benchmark [13]. We normalized the BoW features by whitening the features by their mean and standard deviation computed over the starting training subset."

Below is a demo to normalize features (what I understand from paper):

```
1 load('features.mat');
2 % class_num = 100, train_num = 500, valid_num = 50, test_num = 100
3 init_class_num = 50; % use 50 classes to start training
4 train_x_50 = train_x(:, 1: 50 * train_num);
5 train_y_50 = train_y(1: 50 * train_num);
6 % normalize starting training features
7 train_x_50_mean = mean(train_x_50, 2);
8 train_x_50_std = std(train_x_50, 1, 2);
9 train_x_50_normalized = (train_x_50 - repmat(train_x_50_mean, 1, 50 *
10 train_num)) ./ repmat(train_x_50_std, 1, 50 * train_num);
```

