

# About

---

This project experiments with the 11k Hands dataset by Mahmoud Afifi. We use Color Moments, SIFT, LBP and HOG as feature models and implement distance functions to compare images and people. To combat the dimensionality curse, we explore the effects of feature reduction techniques such as SVD, PCA, LDA and NMF on the above mentioned feature models.

## Requirements:

---

1. Python 3.7.4
2. MongoDB
3. lib32-glib2 to run SIFT executable from <http://www.cs.ubc.ca/~lowe/keypoints/>

## Python libraries

1. (optional) Setup a python virtualenv.
2. Just run `pip install -r requirements.txt` The SIFT executable from David Lowe's page takes about 3 seconds to calculate keypoints for one image. If you would like to use OpenCV's implementation, you need to recompile OpenCV from source using the option `OPENCV_ENABLE_NONFREE`. To do that, clone the `opencv` and `opencv contrib` repositories. Then run

```
$ cmake <path to opencv clone> \  
-D CMAKE_BUILD_TYPE=RELEASE \  
-D BUILD_PYTHON_SUPPORT=ON \  
-D WITH_XINE=ON \  
-D WITH_OPENGL=ON \  
-D WITH_TBB=ON \  
-D BUILD_EXAMPLES=ON \  
-D BUILD_NEW_PYTHON_SUPPORT=ON \  
-D WITH_V4L=ON \  
-D CMAKE_INSTALL_PREFIX=<install path> \  
-D OPENCV_EXTRA_MODULES_PATH=<opencv_contrib clone path> \  
-D OPENCV_ENABLE_NONFREE=ON
```

If you specify `CMAKE_INSTALL_PREFIX`, you need to export this install path to `PYTHONPATH` using `export PYTHONPATH=~/.opencv/lib/python3.7/site-packages/`.

## Configuration

---

The configuration can be edited in `settings.toml`. There are comments in the file describing each of the configurations. To get this program running, it is enough to provide `username`, `password`, `host`, `port`, `data_path`. By default, you need to create folder structure as below. Place the metadata csv into the data folder.

```
|— Data
|   |— <Images dataset>
|— Project Directory
|   |— <Project execution files>
|— Outputs
|   |— <All outputs will be stored here>
```

If you would like the structure to be different, edit the `settings.toml` file to provide the locations of the Data (hand images), Metadata, Output.

## Prerequisite to run any of the tasks

---

1. Generate the metadata for all the images by running `$python db_make.py`.
2. Generate the feature models for all the images in the data directory using `$python db_make.py -m <model>` where `model` can take values `sift|moment|moment_inv|lbp|hog`.

## Tasks interface and execution

---

Every task has its own set of options passed as command line args and are listed below.

### Task 1

This task comprises of 4\*4 variations, each of the feature extraction models can be used and each of the dimensionality reduction techniques can be implemented to obtain k latent semantics.

```
$python task1.py

-m
  Feature model to use. ['sift', 'moment', 'moment_inv', 'lbp', 'hog']
-k
  Number of latent semantics to use.
-frt
  Feature reduction technique to use. ['pca', 'svd', 'nmf', 'lda']
```

### Task 2

This task comprises of 4\*4 variations, each of the feature extraction models can be used and each of the dimensionality reduction techniques can be implemented to obtain 'k' latent semantics and given an image, we found out the 'm' most similar images to the given image.

```
$python task2.py
```

```
-m
    Feature model to use. ['sift', 'moment', 'moment_inv', 'lbp', 'hog']
-k
    Number of latent semantics to use.
-frt
    Feature reduction technique to use. ['pca', 'svd', 'nmf', 'lda']
-n
    Number of top similar images to retrieve
-i
    Name of the query image.
```

## Task 3

This task comprises of 44 variations, each of the feature extraction models can be used and each of the dimensionality reduction techniques can be implemented to obtain 'k' latent semantics for the given label which classifies the image in a particular type, there are a total of  $44 \times 8 = 128$  possible combinations.

```
$python task3.py
```

```
-m
    Feature model to use. ['sift', 'moment', 'moment_inv', 'lbp', 'hog']
-k
    Number of latent semantics to use.
-frt
    Feature reduction technique to use. ['pca', 'svd', 'nmf', 'lda']
-l
    Image label.
    ['male', 'female', 'left', 'right', 'with_acs', 'without_acs', 'dorsal', 'palmar']
```

## Task 4

This task comprises of 4\*4 variations, each of the feature extraction models can be used and each of the dimensionality reduction techniques can be implemented to obtain 'k' latent semantics and given an image, we found out the 'm' most similar images to the given image provided we have picked up images of one particular 'label' provided as input.

```
$python task4.py
```

```
-m
    Feature model to use. ['sift', 'moment', 'moment_inv', 'lbp', 'hog']
-k
    Number of latent semantics to use.
-frt
    Feature reduction technique to use. ['pca', 'svd', 'nmf', 'lda']
-l
```

```
Image label.  
['male','female','left','right','with_acs','without_acs','dorsal','palmar'  
]  
-n  
    Number of similar images required corresponding to the query image.  
-i  
    Query image name.
```

## Task 5

This task comprises of 4\*4 variations, each of the feature extraction models can be used and each of the dimensionality reduction techniques can be implemented to obtain 'k' latent semantics for the given 'label' of images. The task at hand is to use some approach to be able to decide whether an unseen image has the same given label or the opposite label. We have four pairs of possible labels.

1. Right or Left
2. Dorsal or Palmar
3. Male or Female
4. With Accessories or Without Accessories

```
$python task5.py
```

```
-m Feature model to use.['sift','moment','moment_inv','lbp','hog'] -k Number of latent semantics to use. -  
frt Feature reduction technique to use. ['pca','svd','nmf','lda'] -l Image label.  
['male','female','left','right','with_acs','without_acs','dorsal','palmar'] -i Query image name.
```

## Task 6

In this task, we are given a subject ID and the program needs to identify the 3 closest subjects in the given dataset.

```
$python task6.py  
  
-s  
    Query Subject ID.
```

## Task 7

In this task we are given the value of 'k'. We created a subject subject similarity matrix and then applied Non Negative Matrix Factorisation (NMF) on it and display the top 'k' latent semantics which were presented in the form a subject weight pairs, ordered in decreasing order of weights.

```
$python task7.py
```

```
-k  
  K latent semantics.
```

## Task 8

In this task, we created a binary image-metadata matrix and then performed NMF on it. The top 'k' latent semantics in the image space and metadata space were obtained and presented in decreasing order of weights.

```
$python task8.py  
  
-k  
  K latent semantics.
```