

# Instructions for using the CSU Baseline Algorithms

David Bolme and Ross Beveridge  
Computer Science Department  
Colorado State University  
<http://www.cs.colostate.edu/facerec>  
Last Update - June 14, 2012

## Overview

This software is an open source face recognition project being released in the September of 2011. The project will include a new set of face baseline algorithms and also software for computing lighting, focus, and other face quality measures. Currently there are two algorithms:

Local Region Principal Components Analysis (LRPCA) - Released July 2010

Cohort Linear Discriminant Analysis (CohortLDA) - released October 2011

These algorithms are being released for use with NIST's Good, Bad, and Ugly (GBU) challenge problem, although they will of course work on other data sets.

This distribution includes code for running and scoring the algorithms on the Good, Bad and Ugly face recognition data sets. This code is setup to run on Mac OS, however it should be possible to run the code on other platforms if the dependencies are installed correctly. For that, some experience with Python programming is required.

## Release List

New releases of this software will be indexed by date. Here are some of the major changes in previous releases.

- |           |  |
|-----------|--|
| 10/4/2011 | Includes the new CohortLDA algorithm and executables have been updated to conform to the BEST protocol. The distribution also includes trained algorithm configurations. |
| 3/22/2011 | Fixed a minor distance matrix formatting issues on 64bit machines and .  |
| 8/15/2011 | Initial release including the LRPCA algorithm.   |

## Additional Documentation

Automatically generated documentation for the python source code can be produced by ePyDoc using the build\_doc.sh script. This documentation can be viewed by loading

the file doc/html/index.html and describes the python source code.

The file "doc/IARPA-BEST2010.pdf" describes the file formats used by the baseline algorithm. This includes many BEE file formats including sigsets, similarity matrices, and mask matrices.

## Installation of Dependencies

The algorithms should work with Python 2.6 or later and require a few open source Python modules. Below are the dependencies required for the baseline algorithms with the versions with which the algorithm was tested:

Python 2.6.1

Numpy/Scipy

OpenCV 2.2

PIL 1.1.7

PyVision (Included with the distribution)

### Installation on MacOS

As of the most current release we recommend using MacPorts for installation on MacOS, which will keep the dependencies up-to-date. To get MacPorts follow the instructions here:

<http://www.macports.org/install.php>

The instructions here follow:

[http://sourceforge.net/apps/mediawiki/pyvision/index.php?title=Installation\\_MacOS\\_X\\_Snow\\_Leopard#Installing\\_with\\_Python2.6\\_.2F\\_MacPorts\\_.2F\\_64bit](http://sourceforge.net/apps/mediawiki/pyvision/index.php?title=Installation_MacOS_X_Snow_Leopard#Installing_with_Python2.6_.2F_MacPorts_.2F_64bit)

First install Python 2.7

```
$ sudo port install python27
```

To set the new python as the default run the command...

```
$ sudo port install python_select
```

```
$ sudo python_select python27
```

Install PIL, NumPy, and Scipy. (These depend on the numerical library ATLAS which may take many hours to compile.)

```
$ sudo port install py27-pil py27-numpy py27-scipy
```

Install OpenCV with python bindings.

```
$ sudo port install opencv +python27
```

Download and unzip the CSU Baseline Release from here:

```
http://www.cs.colostate.edu/facerec/algorithms/lrpca2010.php
```

Update the PYTHONPATH environment variable to point at the src directory.

```
$ export PYTHONPATH=<PATH_TO_BASELINE>/PythonFaceEvaluation/src/
```

Finally, you may want to install R to create ROC plots.

## Windows Installation

For windows the python components needed to run the baseline algorithm will be installed using binary packages. The following steps describe the packages that need to be installed and the order in which to install them.

Install Python 2.7 (to C:\Python27):

```
http://python.org/ftp/python/2.7.2/python-2.7.2.msi
```

Download and install OpenCV from this URL:

```
http://sourceforge.net/projects/opencvlibrary/files/opencv-win/2.3.1/OpenCV-2.3.1-win-superpack.exe/download
```

Move the directory to C:\OpenCV

Copy c:\OpenCV\build\python\27\cv.\* to c:\Python27\lib\site-packages

Install NumPy from this URL:

```
http://sourceforge.net/projects/numpy/files/NumPy/1.6.1/numpy-1.6.1-win32-superpack-python2.7.exe/download
```

Install SciPy from this URL:

```
http://sourceforge.net/projects/scipy/files/scipy/0.9.0/scipy-0.9.0-win32-superpack-python2.7.exe/download
```

Install Python Imaging Library from this URL (This version has a correction for a font loading problem found in pythonware distribution):

```
http://www.lfd.uci.edu/~gohlke/pythonlibs/ayroe8qu/PIL-1.1.7.win32-py2.7.exe
```

Download the CSUBaselineSystem from this URL:

```
http://www.cs.colostate.edu/facerec/algorithms/lrpca2010.php
```

Unpack the zip file and move the directory to C:\PythonFaceEvaluation

The following paths need to be set up in your environment variables. After adding the variables make sure to close any command prompt windows so that any new command windows get the new variables.

Add C:\Python27 to path

Add C:\opencv\build\x86\vc10\bin to path (for the DLL files)

Create a new environment variable PYTHONPATH=C:\PythonFaceEvaluation\src

Run python from command line and try these commands to make sure all the libraries can be loaded.

```
import numpy
import scipy
import PIL
import cv
import pyvision
```

These should all run without error. No output is expected.

You should now be able to run the executable scripts in the bin directory using the command `python bin/script_name.py`

### **Windows Notes:**

Windows does not come with the bash scripting language that some of the testing scripts require. This should be available through cygwin. Many path names are in unix format so they may need to be updated to run this software on windows.

There is apparently a Font missing with the Python Imaging Library distribution from pythonware that will cause the import of `pyvision.Plot` to fail. This can be fixed by using an alternate PIL, the `PIL-1.1.7.win32-py2.7.exe` from <http://www.lfd.uci.edu/~gohlke/pythonlibs/>.

### **Linux Installation:**

Due to the differences in linux vendors and how software packages are installed, linux is not supported. All of the software is available for linux and the CSU evaluation systems has been run on both Ubuntu and Fedora. Binary installers for dependencies like `opencv`, `PIL`, `Numpy` and `Scipy` are available on most linux distributions. We would suggest using the MacOS installation instructions as a guide for getting things up and running on linux although the process will vary on different linux distributions.

### **Parallels VM:**

For those that want to get up and running quickly there is a Parallel's virtual machine available for download that includes Ubuntu 11.04 and a tested installation of `pyvision` and the full CSU Evaluation System including many other helpful packages such as `Eclipse`, `iPython`, etc.

## About the Executable Scripts

One of the major changes in the recent release is that executable python scripts that conform to the BEST calling protocol have been added to the bin directory. There are three scripts for each algorithm corresponding to automatic, manual, and reduced depending on how the eyes are located. Automatic is a version that does automatic face and eye detection on the source images. The manual configuration reads manually selected eye coordinates from a CSV file. By default these eye coordinates come from the file `sigsets/alleyes.csv`. The reduced version uses small images that have are geometrically normalized to be 512x512 pixels with the eyes located at (192,236) and (320,236) and the filename corresponds to the Recording ID of the image.

## Testing the Installation

The distribution comes with a small subset of the Labeled Faces in the Wild (LFW) dataset that is used to test the code. If everything is set up correctly it should be possible to call a test script that will run the algorithms on the LFW data and produce example output. This script should serve as a good example of how to run the algorithms on new datasets.

```
$ scripts/lfw_test.sh
```

The plot can be found in: `results/lfw_test_100_roc.pdf`

## Running the Algorithms

The first step is to obtain a copy of the MBGC Still Challenge dataset. The process requires you to contact NIST and sign agreements to obtain the data. You will be given an account so that you can download the data through rsync. Once this is downloaded there should be a directory named "StillChallenge" which contains the images for the GBU challenge. Further instructions for obtaining the MBGC dataset are at:

```
http://face.nist.gov/mbgc/
```

The rest of this tutorial assumes that you are running from the command line in the PythonFaceEvaluation directory.

```
$ cd PythonFaceEvaluation
```

The LRPCA and CohortLDA can be run on GBU using this script:

```
$ scripts/gbu_test_manual.sh
```

The script will generate similarity matrices for each partition of the GBU, ROC data, histogram data, and pdf plots of the ROCs. In order to run this test you will need to edit

the script and set the `IMAGE_DIR` variable to point at the top level of the GBU images. (Note: prior releases would scan the directory structure to find images. The new interface requires images to be in a directory structure consistent with the names in the sigsets. There is a script in the bin directory called `collect_images.py` that can be used to help create this directory structure if it does not exist.) If everything executes correctly the following files should have been created:

### **Similarity Matrices:**

```
matrices/cohortlda_manual/matrices/cohortlda_gbu_bad_manual.mtx
matrices/cohortlda_manual/matrices/cohortlda_gbu_good_manual.mtx
matrices/cohortlda_manual/matrices/cohortlda_gbu_ugly_manual.mtx
matrices/lrpca_manual/matrices/lrpca_gbu_bad_manual.mtx
matrices/lrpca_manual/matrices/lrpca_gbu_good_manual.mtx
matrices/lrpca_manual/matrices/lrpca_gbu_ugly_manual.mtx
```

### **Score histogram data:**

```
results/cohortlda_gbu_bad_manual_hist.csv
results/cohortlda_gbu_good_manual_hist.csv
results/cohortlda_gbu_ugly_manual_hist.csv
results/lrpca_gbu_bad_manual_hist.csv
results/lrpca_gbu_good_manual_hist.csv
results/lrpca_gbu_ugly_manual_hist.csv
```

### **ROC data:**

```
results/cohortlda_gbu_bad_manual_roc.csv
results/cohortlda_gbu_good_manual_roc.csv
results/cohortlda_gbu_ugly_manual_roc.csv
results/lrpca_gbu_bad_manual_roc.csv
results/lrpca_gbu_good_manual_roc.csv
results/lrpca_gbu_ugly_manual_roc.csv
```

### **ROC plots:**

```
results/cohortlda_gbu_manual_roc.pdf
results/lrpca_gbu_manual_roc.pdf
```

## **Testing on Other Datasets**

The algorithm testing code can be run on other datasets. For this you will need a sigsets that determine the images that will be used for testing. A two sigsets need to be provided which describe the query and target image sets. Descriptions of the sigset file format can be found in the BEST documentation in the doc directory.

Eye coordinates are needed for the algorithms to run. These can be provided using a CSV format file with 5 columns where the columns are filename, eye1x, eye1y, eye2x, eye2y. To use a new eye coordinates either append them to the end of the sigsets/alleyes.csv file, or modify the `EYE_COORDINATES_FILE` variable in `bin/cohortlda_manual.py` and `bin/lrpca_manual.py` to point to the new

location. If eye coordinates do not exist for the dataset you could try running the algorithms in automatic detection mode by using `bin/cohortlda_automatic.py` or `bin/lrpca_automatic.py`.

## Citing LRPCA and CohortLDA

To use LRPCA, please cite:P.J. Phillips, J.R. Beveridge, B. Draper, G. Givens, A. O'Toole, D. Bolme, J. Dunlop, Y.M. Lui, H. Sahibzada, and S. Weimer, "An introduction to the good, the bad, & the ugly face recognition challenge problem", Image and Vision Computing, vol. 30, no. 3, pp. 206-216, 2012.To use CohortLDA, please cite:Y.M. Lui, D. Bolme, J.P. Phillips, J.R. Beveridge, and B. Draper, "Preliminary Studies on on the Good, the Bad, and the Ugly Face Recognition Challenge Problem", CVPR Biometrics Workspop, Rhode Island, 2012.

## Acknowledgments:

Lead Developer	David Bolme
Additional Development/Algorithms	Ross Beveridge Yui Man Lui Bruce Draper Hao Zhang Nayeem Teli
PyVision Development	Steve O'Hara
Windows Installation Instructions	James R. Matey

Special thanks to the open source community for contributing high quality software with out which these algorithms would not be possible. In particular, those who have contributed to these projects:

- OpenCV
- Python
- Numpy
- Scipy
- Python Image Library