The Journal of Open Source Software

pfla: A Python Package for Dental Facial Analysis using Computer Vision and Statistical Shape Analysis

Maxime Rousseau[1] **and Jean-Marc Retrouvey**[1]

1 McGill University, Faculty of Dentistry **2** Brittle Bone Disease Consortium

DOI: 10.21105/joss.00822

Software

- Review
- Repository
- Archive

Submitted: 12 July 2018
Published: 13 July 2018

Summary

This paper outlines the workings of the software used to conduct automatic facial analyses on patients of the BBDC 7701 protocol investigating the natural history of Osteogenesis Imperfecta.

One of the main advantages of using this software for clinical image analysis is that it automatically places landmarks, standardizing and expediting measurement procedures. The program was written in python and R (R Core Team 2013) programming languages using OpenCV (Itseez 2017) and Dlib (King 2009) libraries, as well as the publicly available facial annotation tool by (Sagonas et al. 2013).

This paper simultaneously presents a broad framework for facial analysis, while describing a specific package applying trained models and statistical analysis. These algorithms can be interchanged (i.e. YOLO (Redmon et al. 2016)) and the R script modified to suit the needs of a particular study.

The program takes as input two folders of dental anteroposterior .jpg images, before assigning landmarks to each object of interest. All coordinates are stored in separate matrices for each group being compared. They are then tested using statistical shape analysis for differences in shape and other attributes of interest to the researcher.

The ___ init___.py file comprises the main method calls, while the classes are stored in the fcn/ directory. Under this directory can be found: img_prep.py, which prepares images by rescaling and converting to grayscale; face_detect.py, which runs the Haar cascade detecting the face on the prepared image; annotate.py, which places landmarks on the detected faces; and analyze.py, which calls the stats.R script to perform statistical analyses.

The program stores output images in their respective directories: img_raw/ for raw inputted images, img_prep/ for prepared images, and img_proc/ for processed images (face bound by a rectangle with landmarks placed).

The data/ directory contains the cascade classifier and shape predictor. Under faces/ are stored coordinates of the rectangles from the detected faces in each image. The ldmks/ directory contains the matrices of landmarks for each group to be analyzed using the R script.

After initial preparation, images go through a Haar Cascade classifier trained to detect faces (Viola and Jones 2001). This algorithm scans the input through the scope of a small rectangle. It sums up the mean features thus detected, comparing them to sections of the face training set. For our purpose, the algorithm was trained on faces, hence it may recognize facial features such as eyes, noses, etc. This allows us to draw a bounding box around the face detected in the input image. Once initial image processing is completed,

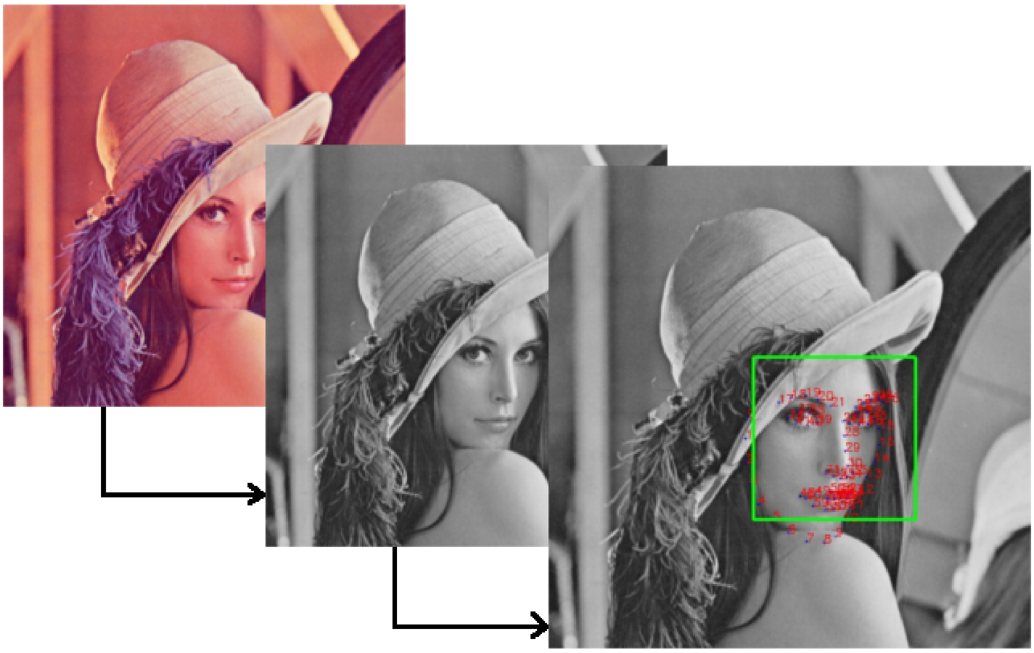

Figure 1: Image Processing Example Over the Famous Lena Image

a landmark template is applied which produces a matrix of 68 (x, y) coordinates for each patient. The outputted matrices help detect facial manifestations of disease by comparing groups of patients with clinical conditions to controls.

The antero posterior analysis consists of $l = 68$ landmarks automatically placed on patient photographs via software. These sets of coordinates produce matrices of $k = 2$ dimensions. The matrices are represented as such:

$$M_{patient} = [x_1, x_2, ..., x_l, y_1, y_2, ..., y_l]$$

Where l represents the number of points attached to a photographs.

Statistical shape analysis has mostly been used in the field of evolutionary biology for the analysis of skeletal artifacts. It also has applications in the medical fields, most notably in imaging analysis. The pfla package performs the image processing described above in order to conduct statistical analysis. The R script uses the “shapes” package by (Dryden 2017). First, the various matrices produced by our data are aligned. This is done by performing a Generalized Procruste Analysis (GPA). This allows for shapes matched in proportion and orientation. This is needed for the purpose of the study, given the authors’ interest in morphological differences. The algorithm operates as follows:

1. arbitrarily choose a reference shape (usually from available instances)
2. superimpose all instances to current reference shape
3. compute mean shape of the current set of superimposed shapes
4. if the Procruste distance between the mean shape and the reference shape is above a given threshold, set reference to mean shape and reiterate from step 2

Once our matrices are aligned, they are transformed into unidimensional matrices through orthogonal projection by performing a Principal Component Analysis (PCA). This highlights features present in the dataset in order to facilitate comparison between groups. The vectors produced by the linearization of our datasets will be annotated as such:

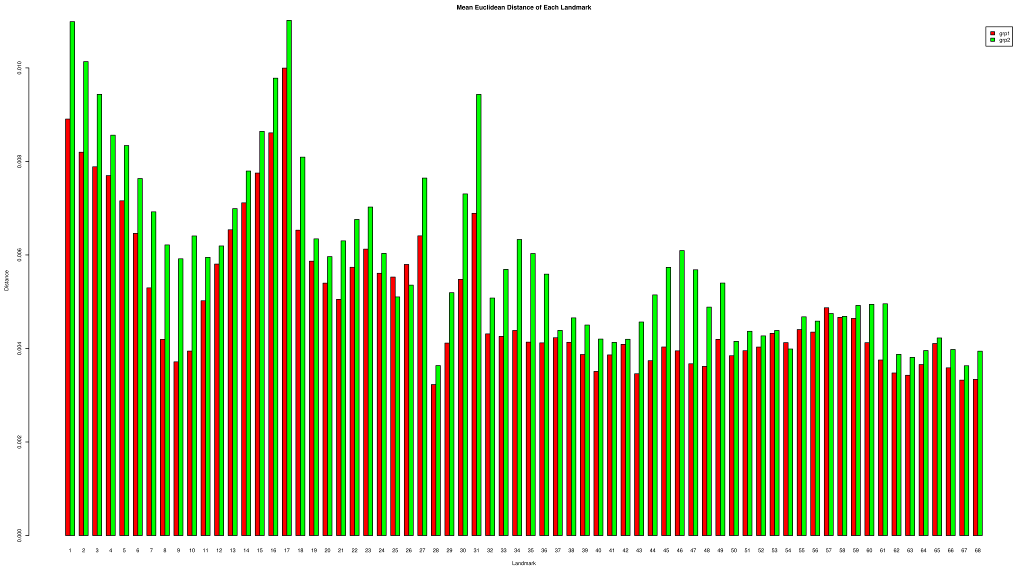

Figure 2: Mean Euclidean Distance Output Histogram

$$V_{patient} = [i_1, i_2, ..., i_{2l}]$$

Following the PCA, the Goodall F test is computed on the mean shapes of each group using the non-parametric Bootstrap method to compare multivariate matrices (Brombin and Salmaso 2009). It is unreasonable to assume isotropy as well as equal covariance between the matrices being studied. This can be explained by the simple fact that human faces are naturally asymmetrical. After testing our hypothesis, exploration of the data is continued in the hope of finding an explanation for these results. This is accomplished by computing the mean Euclidean distance of each landmark from its corresponding landmark on the baseline shape. Given values for each landmark in a particular group are then compiled, resulting in a set of mean distances per landmark from baseline. This allows for interpretion the results on a deeper level, isolating the greatest differences and similarities between study subjects. Mean and standard deviation of a particular set of landmarks can therefore be assessed more broadly.

It is important to understand that this is a morphological analysis, hence only relative shape is evaluated. Conclusions related to size cannot be drawn from this method.

Accuracy of the image processing can be visualized by inspecting detected faces and landmarks (Figure 1). The program outputs a histogram of mean Euclidean distances from the baseline for each group (Figure 2).

References

Brombin, Chiara, and Luigi Salmaso. 2009. "Multi-Aspect Permutation Tests in Shape Analysis with Small Sample Size" 53 (12):3921–31. https://doi.org/https://doi.org/10.1016/j.csda.2009.05.010.

Dryden, I. L. 2017. *shapes Package*. Vienna, Austria: R Foundation for Statistical Computing. http://www.R-project.org.

Itseez. 2017. "Open Source Computer Vision Library." https://github.com/itseez/opencv.

King, Davis E. 2009. "Dlib-Ml: A Machine Learning Toolkit." *Journal of Machine Learning Research* 10:1755–8.

R Core Team. 2013. *R: A Language and Environment for Statistical Computing*. Vienna, Austria: R Foundation for Statistical Computing. http://www.R-project.org/.

Redmon, J., S. Divvala, R. Girshick, and A. Farhadi. 2016. "You Only Look Once: Unified, Real-Time Object Detection." In *2016 Ieee Conference on Computer Vision and Pattern Recognition (Cvpr)*, 779–88. https://doi.org/10.1109/CVPR.2016.91.

Sagonas, C., G. Tzimiropoulos, S. Zafeiriou, and M. Pantic. 2013. "A Semi-Automatic Methodology for Facial Landmark Annotation," June, 896–903. https://doi.org/10.1109/CVPRW.2013.132.

Viola, P., and M. Jones. 2001. "Rapid Object Detection Using a Boosted Cascade of Simple Features." In *Proceedings of the 2001 Ieee Computer Society Conference on Computer Vision and Pattern Recognition. CVPR 2001*, 1:I–511–I–518vol.1. https://doi.org/10.1109/CVPR.2001.990517.