

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

Maxime Rousseau<sup>1</sup> and Jean-Marc Retrouvey<sup>1</sup>

<sup>1</sup> McGill University, Faculty of Dentistry <sup>2</sup> Brittle Bone Disease Consortium

DOI: [10.21105/joss.00855](https://doi.org/10.21105/joss.00855)

## Software

- [Review](#) ↗
- [Repository](#) ↗
- [Archive](#) ↗

Submitted: 12 July 2018

Published: 02 December 2018

## License

Authors of papers retain copyright and release the work under a Creative Commons Attribution 4.0 International License ([CC-BY](#)).

## Introduction

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 (OI).

The software takes two image directories containing facial pictures of the study subjects as inputs. Each image is processed through face detection and landmark allocation producing a matrix of coordinates for each study group. These matrices are then used for the statistical comparison of the groups. The results from the analysis is printed to the terminal (Figure 1).

One of the main advantages of using this software for clinical image analysis is that it automatically places landmarks, standardizing and expediting measurement procedures. This task was previously conducted manually which was very time consuming and prone to operator error and bias. With this novel approach we are able to circumvent these issues allowing us to have an efficient and objective way of conducting facial analysis.

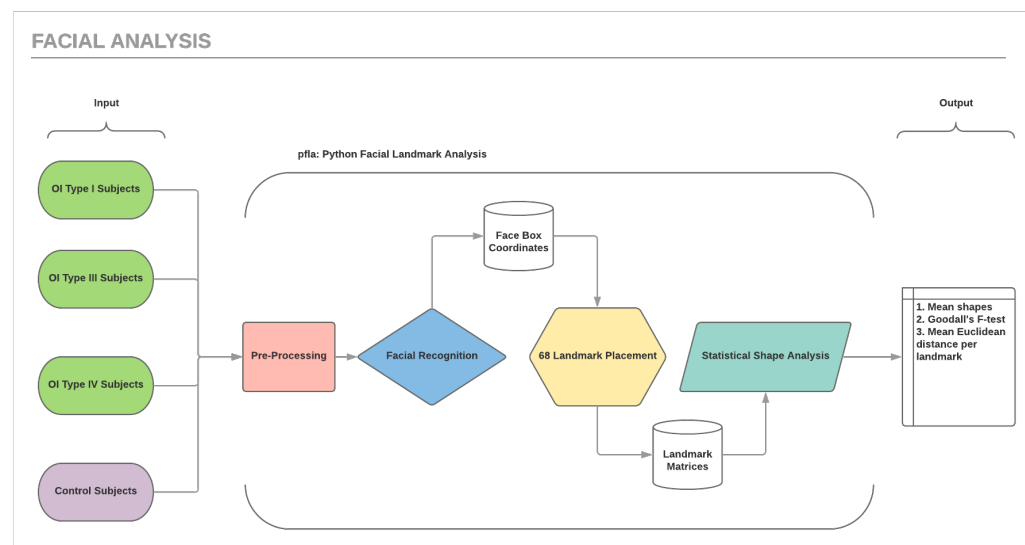
This paper simultaneously presents a broad framework for facial analysis, while describing a specific package applying trained models and statistical analysis. These algorithms can be interchanged (i.e. YOLO (Redmon, Divvala, Girshick, & Farhadi, 2016)) and the R script modified to suit the needs of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

## Software

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, Tzimiropoulos, Zafeiriou, & Pantic, 2013).

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.



**Figure 1:** Schematic representation of the pfla software for OI facial analysis

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.

## Image Processing

After initial preparation, images go through a Haar Cascade classifier trained to detect faces (Viola & 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, 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 matrix for a single study group are represented as such:

$$m^{[s]} = [x_1, x_2, \dots, x_{68}, y_1, y_2, \dots, y_{68}]$$

Where the  $s$  superscript represents the subject attached to this row of coordinates.

## Statistical Shape Analysis

Statistical shape analysis has mostly been used in the field of evolutionary biology for the analysis of skeletal artifacts (I. L. Dryden, 2017). 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 (I. L. Dryden & Mardia, 1992). The R script uses the “shapes” package by (Ian L Dryden & Mardia, 2016). First, the various matrices produced by our data are aligned. This is done by performing a Generalized Procrustes Analysis (GPA)(Gower, 1975),(Ten Berge, 1977). 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 Procrustes distance between the mean shape and the reference shape is above a given threshold, set reference to mean shape and reiterate from step 2

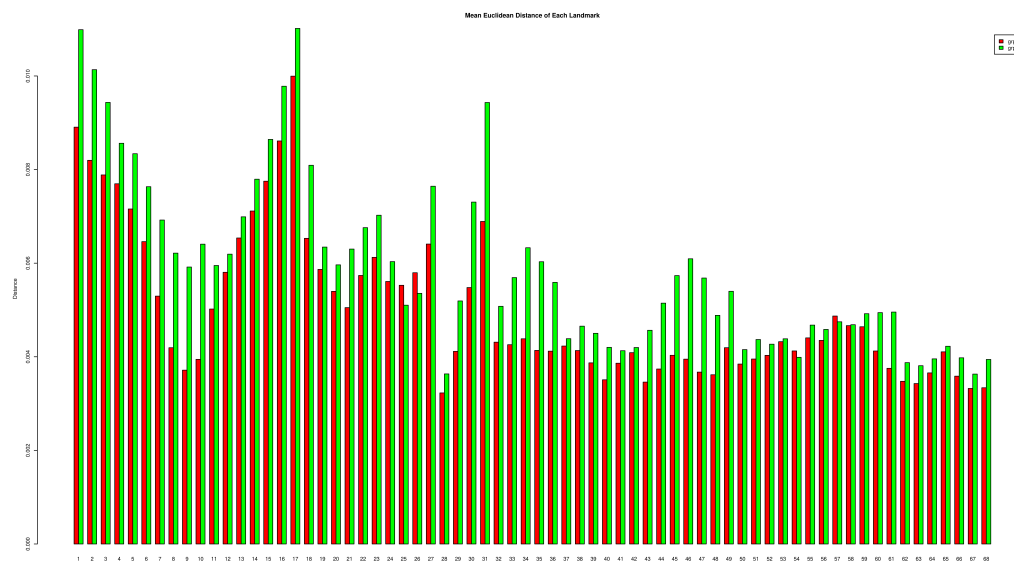
Once our matrices are aligned, they are transformed into unidimensional matrices through orthogonal projection by performing a Principal Component Analysis (PCA). This highlights features present in the dataset in order to facilitate comparison between groups. The vectors produced by the linearization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

$$v^{[s]} = [i_1, i_2, \dots, i_{136}]$$

Following the PCA, the Goodall F test (Goodall, 1991) is computed on the mean shapes of each group using the non-parametric Bootstrap method to compare multivariate matrices (Brombin & 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 interpretation 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. The program outputs a histogram of mean Euclidean distances from the baseline for each group (Figure 2).



**Figure 2:** Mean Euclidean Distance Output Histogram

## References

- Brombin, C., & Salmaso, L. (2009). Multi-aspect permutation tests in shape analysis with small sample size, *53*(12), 3921–3931. doi:<https://doi.org/10.1016/j.csga.2009.05.010>
- Dryden, I. L. (2017). *shapes package*. Vienna, Austria: R Foundation for Statistical Computing. Retrieved from <https://cran.r-project.org/web/packages/shapes/shapes.pdf>
- Dryden, I. L., & Mardia, K. V. (1992). Size and shape analysis of landmark data. *Biometrika*, *79*(1), 57–68. doi:[10.1093/biomet/79.1.57](https://doi.org/10.1093/biomet/79.1.57)
- Dryden, I. L., & Mardia, K. V. (2016). *Statistical shape analysis: With applications in r*. John Wiley & Sons.
- Goodall, C. (1991). Procrustes methods in the statistical analysis of shape. *Journal of the Royal Statistical Society. Series B (Methodological)*, 285–339.
- Gower, J. C. (1975). Generalized procrustes analysis. *Psychometrika*, *40*(1), 33–51. doi:<https://doi.org/10.1007/BF02291478>
- Itseez. (2017). Open source computer vision library. <https://github.com/itseez/opencv>.
- King, D. E. (2009). Dlib-ml: A machine learning toolkit. *Journal of Machine Learning Research*, *10*, 1755–1758.
- R Core Team. (2013). *R: A language and environment for statistical computing*. Vienna, Austria: R Foundation for Statistical Computing. Retrieved from <http://www.R-project.org/>
- Redmon, J., Divvala, S., Girshick, R., & Farhadi, A. (2016). You only look once: Unified, real-time object detection. In *2016 IEEE conference on computer vision and pattern recognition (cvpr)* (pp. 779–788). doi:[10.1109/CVPR.2016.91](https://doi.org/10.1109/CVPR.2016.91)
- Sagonas, C., Tzimiropoulos, G., Zafeiriou, S., & Pantic, M. (2013). A semi-automatic methodology for facial landmark annotation, 896–903. doi:[10.1109/CVPRW.2013.132](https://doi.org/10.1109/CVPRW.2013.132)
- Ten Berge, J. M. (1977). Orthogonal procrustes rotation for two or more matrices. *Psychometrika*, *42*(2), 267–276. doi:<https://doi.org/10.1007/BF02294053>

Viola, P., & Jones, M. (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* (Vol. 1, pp. I–511–I–518vol.1). doi:[10.1109/CVPR.2001.990517](https://doi.org/10.1109/CVPR.2001.990517)