

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

17 May 2018

## Summary

In this paper we will propose a new methodology for clinical image analysis, the Detection Outline Analysis (DOA) framework. This technique uses statistical shape analysis and computer vision. To demonstrate its efficiency, we will present a python package using this method for automatic analysis antero posterior pictures of dental patients. The code will be made available for the benefits of the scientific community.

The program takes as input two folders of dental anteroposterior .jpg images. Each image it takes as input the object of interest is first detected, then landmarks are assigned to this object. All coordinates for these landmarks are stored in matrices for the different groups being compared. Finally through statistical shape analysis, the groups are tested for the presence of statistical differences in shapes or other analyses of interest to the researcher.

One of the main advantages for clinical image analysis following this method is that the landmarks will be automatically placed by the software, allowing us to standardise the measurement procedures. The computer program written in the python and R(R Core Team 2013) programming languages using the OpenCV(Itseez 2017) and Dlib(King 2009) libraries as well as the publicly available facial annotation tool by (Sagonas et al. 2013) We will then store the coordinates of these landmarks in a two dimensional matrix using a csv type file for later use in our statistical analysis.

As the objective of this paper are both to a package applying specific trained models and statistical analysis and presenting the broader outline of a framework for facial analysis, it is important to understand that algorithms can be interchanged (i.e. YOLO(Redmon et al. 2016)).

The `__init__.py` file comprises of the main method calls while the differ-

ent classes are stored in the `fcn/` directory. Under this directory, we find: `img_prep.py` which will prepare the image by scaling and transforming it to grayscale, `face_detect.py` which runs the haar cascade detecting the face on the prepared image, `annotate.py` which places the landmarks on the detected faces of the image, `analyze.py` calls the `stats.R` script which runs the statistical analyses for the study.

The output images are stored as they are processed in their respective directories: `img_raw/` for the raw inputted images, `img_prep/` for the prepared images, `img_proc/` for the processed images (faces detected and landmarks placed).

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

When processing images, it first scales them to a set size to assure the uniformity of the dataset. It is important to not that are not cropped hence aspect ratios should be similar across the whole dataset of images. These are then transformed to grayscale. After the initial preparation has been executed these images then go through a Haar Cascade classifier which was trained to detect faces[?]. This algorithm functions by scanning the input through the scope of a small rectangle. It sums up the mean features of that said rectangle then compares it to sections of the face training set. For our case, 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 from the input image. Once the initial image processing is completed a landmark template is applied to the detected faces which produces a matrix of 68 (x, y) coordinates for each patients. These are then used to conduct facial analysis on the patient in order to help our diagnostic procedures or treatment planning. We can also use the outputed matrices to compare groups of patients with clinical conditions to help us detect facial manifestations of a disease. The antero posterior analysis will consist of  $l = 68$  landmarks automatically placed on patient photographs by a computer program. These sets of coordinates will produce matrices of  $k = 2$  dimensions. The matrices will be represented as such:

$$M_{patient} = [x_1, x_2, \dots, x_l, y_1, y_2, \dots, y_l]^T$$

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 artifact. It also has applications in the medical fields, most notably in imaging analysis. Using the matrices produced by the process explained above We will be using the matrices generated from the image processing in order to conduct the statistical analysis. First, we must align the various matrices produced by the whole of our data. We will be doing so by doing a Generalized Procruste Analysis (GPA) (cite article). This will allow us to work with shape matched in proportion and orientation. This is needed in our

case given that we are interested 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 have been aligned we will transform them into unidimensional matrices through orthogonal projection by performing a Principal Component Analysis (PCA). This will aid us highlight the features present in the dataset in order to facilitate comparison between groups. The vectors produced by this linearization of our datasets will be annotated as such:

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

Following the PCA, we will be conducting a Goodall F test on the mean shapes of each group using the non-parametric Bootstrap method to compare our multivariate matrices (Brombin and Salmaso 2009). Two reasons explain our choice for this test, it is unreasonable to assume isotropy as well equal covariance between the population matrices being studied. This can be explained by the simple fact that our matrices contain coordinates of human faces which are asymmetrical. After having tested our hypothesis we will continue to explore the data in the hopes of finding an explanation for the results of the hypothesis test. This will be accomplished by computing the mean euclidean distance of each landmark from its corresponding landmark on baseline shape. We then compile the given values for each landmarks in a particular group resulting in a set of mean distances per landmark from baseline. This will allow us interpret the results on a deeper level as we will be able to isolate where are the greatest differences and similarities between the study subjects. We can also assess more broadly the mean and standard deviation of a particular set of landmark.

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

We can visualize the accuracy of the image processing by inspecting the detected face and landmarks (Figure 1). The program also outputs a histogram of the mean euclidean distance from baseline for each group (Figure 2). where we have the female group in red and the male group in green.

Automatic facial landmarking and statistical shape analysis has proved to be a reliable, reproducible and efficient way of conducting facial analysis.

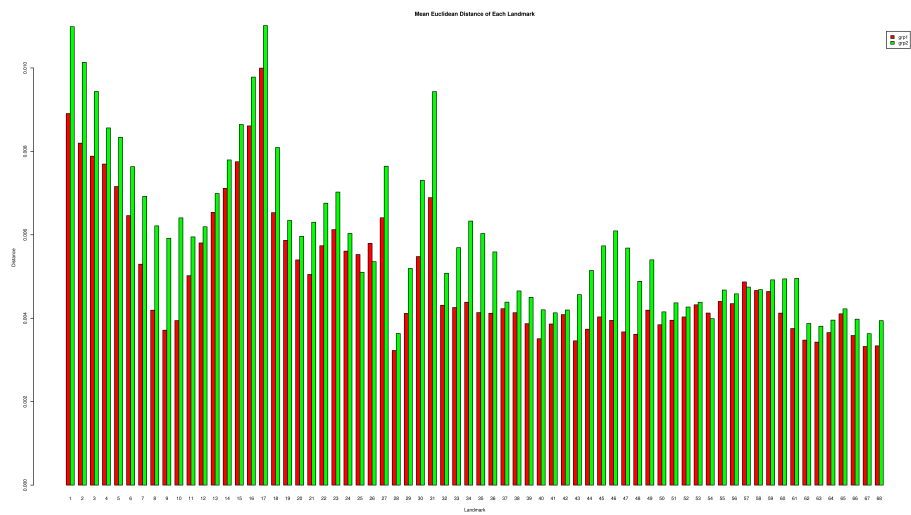


Figure 1: Mean Euclidean Distance Output Histogram

## References

- Brombin, Chiara, and Luigi Salmaso. 2009. “Multi-Aspect Permutation Tests in Shape Analysis with Small Sample Size.” *Computational Statistics & Data Analysis* 53 (12): 3921–31. doi:<https://doi.org/10.1016/j.csda.2009.05.010>.
- 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. doi: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. doi:10.1109/CVPRW.2013.132.