Recognition of Occluded and Lateral Faces Using MTCNN, DLIB and Homographies

Gustavo Alves Bezerra
Departamento de Matemática e
Informática Aplicada
Universidade Federal do Rio Grande do Norte
Natal, Rio Grande do Norte 59078–970
Email: gustavowl@lcc.ufrn.br

Rafael Beserra Gomes
Departamento de Matemática e
Informática Aplicada
Universidade Federal do Rio Grande do Norte
Natal, Rio Grande do Norte 59078–970
Email: rafaelbg@dimap.ufrn.br

***Abstract*—With the advance of technology it is possible to create more robust security systems. For this task, image processing alongside Deep Neural Networks are currently being used in several works for facial recognition. However, occlusions and faces in different angles are a challenge for most algorithms. Attempting to contour this issue, an algorithm for facial recognition combining MTCNN, DLIB and homographies is proposed. In the obtained results, a comparison between the proposed algorithm and basis works indicates that, for some controlled cases, a mean accuracy improvement of 7.4% was obtained, with a maximum of 8.23% for occluded faces and 14.08% for lateral faces.**

I. Introduction

Security is an issue since ancient times. Nowadays, police and cameras are used to achieve security. In addition, aiming a better performance, repetitive tasks that were performed by humans are now executed by robots.

The purpose of this manuscript is to improve facial recognition, including occluded and lateral faces. Currently, it is attempted to detect new people on-the-fly via pattern matching. The faces are detected using Multi-task Cascaded Convolutional Network (MTCNN) developed by Zhang et al. [1]. Most of the facial traits are extracted using a slightly swallower version of the Deep Residual Network by He et al. [2]; which was developed and publicly provided by King, Davis E. The traits are then analyzed and assigned to each person.

The remaining of this manuscript is organized as follows: section II describes related works on facial landmarks detection; section III focuses on briefly explaining the neural networks used; section III-C explains how the algorithm works, and the steps to increase its accuracy and section V shows the results obtained.

II. Related Works

The Menpo Challenge provides a dataset from frontal and lateral faces [3]. However, different landmarks are used for these scenarios. The WIDER FACE dataset's focuses on different poses, illumination and specially scenarios [4].

Yang et al.'s objective is to detect faces in every possible situation [4]. Face recognition is not concerned, though. Yang et al. proposed a CNN that uses ranges of scales for faces alongside divide and conquer [4].

The Annotated Facial Landmarks in the Wild (AFLW) dataset contains $25,993$ landmarked images gathered from Flickr [5]. Also, Koestinger et al. say that "AFLW is well suited to train and test algorithms for multi-view face detection, facial landmark localization and face pose estimation". Analogously, Labeled Facial in the Wild (LFW) has similar purposes in unconstrained environments [6] [7].

Facial recognition in videos is not far from the scope of the problem approached in this paper. An example of such benchmark is available in Shen et al.'s work [8].

III. Deep Neural Networks Used

The Deep Neural Networks (DNN) used were developed for identifying, extracting features and recognizing faces on images. Each DNN can only perform a subset of these tasks.

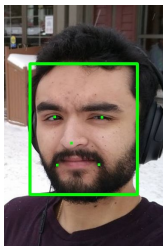
A. MTCNN

The Multi-task Cascaded Convolutional Network (MTCNN) developed by Zhang et al. [1] receives an image as input and outputs seven points for each detected face. The first two points correspond to the limits of the bounding box for a face, i.e. the top left point and the bottom right point. The remaining five correspond to the position of the left eye pupil, right eye pupil, nose tip, left mouth corner and right mouth corner. Zhang et al. claim that a mean error of 6.9% was obtained on AFLW benchmark.

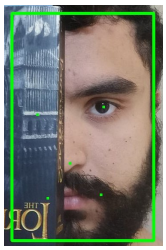
The advantage of MTCNN is its accuracy regarding different poses, illumination and, specially, occlusion. Some simple examples are illustrated in figure 1. Figure 1a shows the output of MTCNN for a face with reasonable position and illumination. Although the facial expression is not neutral, only the mouth corners' positions present some negligible error. It is possible to note in figure 1b that even though half of the face is covered, MTCNN predicted reasonably well the position of the occluded eye and mouth corner.

B. Shape Predictor

The DLIB library was originally developed in C++ with the purpose of supporting machine learning applications [9]. DLIB provides a shape predictor for faces ("shape_predictor_68_face_landmarks") which returns 68



(a) No Occlusion



(b) Occlusion

Fig. 1. Two simple figures processed by MTCNN. The points and the bounding box returned were drawn in the images.

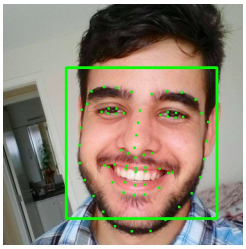


Fig. 2. A simple figure processed by DLIB's shape predictor.

characteristic points for each person given the face bounding box. Using DLIB, the bounding box can be obtained through the face detector (get_frontal_face_detector).

DLIB is very accurate for extracting the points from frontal faces (figure 2). Compared with figure 1a, it is possible to see that MTCNN and DLIB have three points in common, namely the nose tip and mouth corners. However, DLIB extracts the eyes' contour while MTCNN outputs the middle of the eyes.

C. Face Recognition

King, Davis E. [9] has also publicly provided a pre-trained Resnet Network based on He's work [2] [1]. It receives a vector V_1 as input ($V_1 \in (\mathbb{R} \times \mathbb{R})^n, n \in \mathbb{N}$) and outputs a vector [2] $V_2 \in \mathbb{R}^{128}$. Ideally, V_1 should be the vector of 68 points extracted with the shape predictor, i.e. $V_1 \in (\mathbb{R} \times \mathbb{R})^{68}$. The $\mathbb{R}^{128}$ vector space is enough to distinguish between faces. Also, two vectors $V_3, V_4 \in \mathbb{R}^{128}$ represent similar faces if the Euclidean Distance between them is closer to 0.

IV. Face Recognition in Images

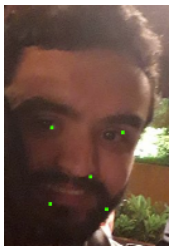
An introductory DLIB algorithm [3] was used as basis for the current implementation. Beforehand, it is necessary to understand some problems with the original approach.

A. Introductory Algorithm

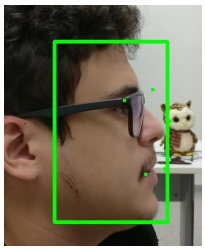
The Introductory Algorithm basically combines the DLIB's Shape Predictor (section III-B) and Face Recognition (section III-C). It processes the image with the Shape Predictor, obtaining a bounding box and the 68 characteristic points for each face. Then the Face Recognition Resnet receives the 68 points and outputs the characteristic vector V_f for that face ($V_f \in \mathbb{R}^{128}$).

The algorithm tries to associate images between two specified folders. One is the database folder, i.e. it contains all known faces. The other is the target folder, i.e. it contains faces that may be unknown and different images of known faces.

First, the algorithm processes the known faces, generating one $\mathbb{R}^{128}$ vector for each face. These vectors will then be used as a type of "basis" for the "face vector space". Second the algorithm will process the potentially unknown faces and attempt to associate them to the known ones. For each new face, a corresponding $\mathbb{R}^{128}$ vector is computed. Then, it is compared with each vector in the face vector space by computing the $norm = vector\_norm(V_{fb} - V_{nf}), where\ V_{fb}, V_{nf} \in \mathbb{R}^{128}$, $V_{fb} \in\ face\ vector\ space\ basis$ and $V_{nf} \in\ new\ faces$, and $\{norm \in \mathbb{R} \mid 0 \leq norm \leq 1\}$. Hence, the closer the value of $norm$ is to zero, the more similar are the compared faces. Lastly, the Introductory Algorithm tries to find a match accordingly to the values of $norm$. A $norm$ is considered a match if, for a given value of $tolerance$, $\{tolerance \in \mathbb{R} \mid 0 \leq tolerance \leq 1\}$, $norm \leq tolerance$. The Introductory Algorithm returns the **first** match it founds.



(a) Unfavorable Illumination



(b) Unfavorable pose

Fig. 3. Two examples with unfavorable conditions where MTCNN outperforms DLIB's Face Detector.

B. MTCNN with Min Match Algorithm

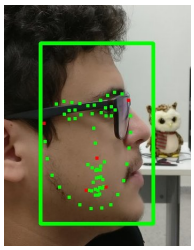
The first problem of the Introductory Algorithm is that it searches for the first match. For instance, if $tolerance = 0.55$ and the vector of norms is $[0.7, 0.54, 0.3]$, then the Introductory Algorithm would output the face corresponding to 0.54 as being the matched face; although the face corresponding to 0.3 is a better match. This issue is easily solvable by simple searching for the **min** match.

Another issue is the use of DLIB's Face Detector to identify faces. MTCNN is more accurate for this task. For instance, Shape Predictor cannot find many occluded faces such as the one identified by MTCNN in figure 1b or faces with unfavorable illumination conditions such as figure 3a. Also, DLIB's Face Detector only identifies frontal faces; while MTCNN can find faces in unfavorable angles (figure 3b).

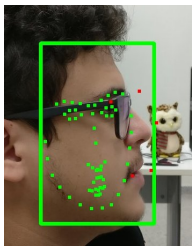
[1] http://blog.dlib.net/2017/02/high-quality-face-recognition-with-deep.html

[2] https://github.com/davisking/dlib/blob/master/examples/dnn_face_recognition_ex.cpp

[3] https://towardsdatascience.com/facial-recognition-using-deep-learning-a74e9059a150

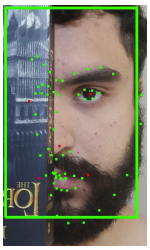


(a) Unfavorable Illumination

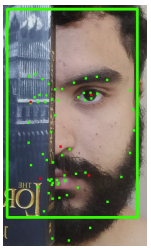


(b) Unfavorable pose

Fig. 4. Extracting facial features using different techniques. The face's bounding box was detected using MTCNN. The points of figure 4a were extracting using DLIB's Shape Predictor only. The points of figure 4b were also extracted with DLIB's Shape Predictor (green points), but some were substituted by MTCNN's points (red ones).



(a) MTCNN and DLIB



(b) Homography

Fig. 5. Comparison between the points extracted by MTCNN and DLIB with the same points after applying a homography having DLIB points as source plane and MTCNN points as destiny plane.

This improved version of the algorithm uses MTCNN instead of DLIB to identify faces. Then, it sends all the detected bounding boxes to the Shape Predictor so it can extract the 68 characteristic points. This introduces a new issue to the problem, since the Shape Predictor expects a frontal face (figure 4a). It is possible to notice, however, that the Shape Predictor is reasonably accurate for the eye and eyebrow positions. In order to diminish this inaccuracy, five Shape Predictor's points are substituted by the five MTCNN points (red points in figures 4a and 4b). The correspondence between DLIB and MTCNN's points are predefined, once each facial landmark data variables (e.g. nose coordinates) are identified beforehand.

C. MTCNN with Homography

A problem rises when treating occluded faces: a scenario similar to figure 5a happens. It is possible to verify that DLIB "chops" the face side occluded by the book. In order to diminish the effects of this error, all Shape Predictor's points are moved toward the MTCNN points based on homography of four or five associated pairs from both.

In summary, given two plans (A and B) in a three dimensional space, a homography maps the points of plan A to plan B through a linear operator.

MTCNN's points are more accurate than DLIB's. Hence, they are used as destination points. In other words, the homography will attempt to map the DLIB's deformed plan to MTCNN's more stable plan. For the homography, either five or four points are used: eyes, mouth corners and optional nose. The nose is optional because, considering the three dimensional space, it does not form a plan with the remaining four points. A result can be seen in figure 5b: the homography stretches the occluded side attempting to fit the real face. It is possible to verify in figure 5b that the green points are surrounding the red ones; which did not happen in figure 5a.

V. Results

The objective of the implemented algorithm was to associate target images with images in a database. The set of target images also contains pictures of people that are not in the database. In therms of association accuracy, the results were not so different between approaches, as can be seen in Table I. This probably happens due to the considerably small size of the database and the high quality of the images. The small size of the database reduces the number of possible tests; while the high quality of the images contribute to a higher accuracy of points detection. Even so, the results using homography were more accurate.

TABLE I
Classification Results

Technique	Correct	Wrong	Ratio (C/W)
DLIB	59	41	1.44
MTCNN	59	41	1.44
5-points	62	38	1.63
4-points	62	38	1.63

In order to compare the different approaches, it is possible to use the difference between the obtained $norms$ as a metric. The $norms$ computed by DLIB are used as basis and the efficiency is calculated as follows:

$$efficiency = wrong + correct \tag{1}$$

where, for every non-expected match i:

$$wrong = \sum_i approach_i - dlib_i \tag{2}$$

and for the expected match j (necessarily, $j \neq i$):

$$correct = dlib_j - approach_j \tag{3}$$

Two different approaches can be compared via the max function.

Table II is obtained by applying these formulas to the result of four chosen images: in database, out of database, occlusion and side.

Hence, one may erroneously conclude that the hybrid MTCNN and DLIB approach is not sufficient to improve the accuracy of face recognition. However, this result depends on the $tolerance$ factor used and on the database. In fact, for some images, the accuracy of the $norms$ was improved by

TABLE II
COMPARISON BETWEEN APPROACHES

Image	mixed	5-points homography	4-points homography
in	-0.00672	0.0938	0.07216
out	-0.04372	-0.08464	-0.07186
occl	-0.00227	0.0283	0.08228
side	0.0068	0.14081	0.04475

TABLE III
LFW SUBSET RESULTS

Technique	Tolerance	Correct	Wrong	Ratio (C/W)
5-points	0.4	163	622	0.26
5-points	0.5	272	513	0.53
5-points	0.6	275	510	0.54
4-points	0.4	134	651	0.21
4-points	0.5	237	548	0.43
4-points	0.6	275	510	0.54

0.02. It may not be a large amount, but that difference was sufficient to diminish the number of false matches without varying the $tolerance$ value. Also, note that 5-points homography is better than DLIB by a similar factor (0.0283).

In addition, when homography is applied, it raises the accuracy considerably. It is not possible to conclude, however, if the 5-points homography is better than 4-points. Apparently, 5-points is more precise for lateral faces while 4-points homography performs better for occluded faces. It is necessary to test these two techniques with more images to conclude which would give more satisfactory results in the general case.

However, executing the algorithm for a subset of the LFW dataset [6] [7] containing 785 images, the result of table III was obtained. From this, it is possible to see that although applying homographies generated satisfactory results for a controlled dataset, it is necessary to refine the technique so the changes generated by the homographies do not prejudice DLIB and MTCNN's accuracy.

VI. CONCLUSION

The result achieved was very satisfactory for a limited image database. Although there are many accurate algorithms for facial recognition, this work focuses on improving recognition of occluded and lateral faces. For frontal faces, it was expected that the result would be very similar to the basis works. For the occluded and lateral faces cases, accuracy was satisfactorily improved, though.

On a larger dataset, however, it was observed that using homographies inserted some noise that harmed the algorithm precision. Future works may focus on reducing the homographies' negative influence.

With a few changes, this algorithm may also perform well for face recognition in videos. Although it is necessary to improve its performance so it can be used in real time applications.

REFERENCES

[1] K. Zhang, Z. Zhang, Z. Li, and Y. Qiao, "Joint face detection and alignment using multitask cascaded convolutional networks," *IEEE Signal Processing Letters*, vol. 23, no. 10, pp. 1499–1503, 2016.

[2] K. He, X. Zhang, S. Ren, and J. Sun, "Deep residual learning for image recognition," in *Proceedings of the IEEE conference on computer vision and pattern recognition*, 2016, pp. 770–778.

[3] S. Zafeiriou, G. Trigeorgis, G. Chrysos, J. Deng, and J. Shen, "The menpo facial landmark localisation challenge: A step towards the solution," 2017, pp. 2116–2125.

[4] S. Yang, P. Luo, C. Loy, and X. Tang, "Wider face: A face detection benchmark," 2016, pp. 5525–5533.

[5] M. Koestinger, P. Wohlhart, P. M. Roth, and H. Bischof, "Annotated facial landmarks in the wild: A large-scale, real-world database for facial landmark localization," in *Computer Vision Workshops (ICCV Workshops), 2011 IEEE International Conference on*. IEEE, 2011, pp. 2144–2151.

[6] G. B. Huang, M. Ramesh, T. Berg, and E. Learned-Miller, "Labeled faces in the wild: A database for studying face recognition in unconstrained environments," University of Massachusetts, Amherst, Tech. Rep. 07-49, October 2007.

[7] G. B. H. E. Learned-Miller, "Labeled faces in the wild: Updates and new reporting procedures," University of Massachusetts, Amherst, Tech. Rep. UM-CS-2014-003, May 2014.

[8] J. Shen, S. Zafeiriou, G. Chrysos, J. Kossaifi, G. Tzimiropoulos, and M. Pantic, "The first facial landmark tracking in-the-wild challenge: Benchmark and results," 2015, pp. 1003–1011.

[9] D. E. King, "Dlib-ml: A machine learning toolkit," *Journal of Machine Learning Research*, vol. 10, no. Jul, pp. 1755–1758, 2009.