Noname manuscript No.
(will be inserted by the editor)

Human-Machine Interaction based on Eye Command:

Algorithm to detect the movement of the eyes and recognize the command

Saliha Benkerzaz · Youssef Elmir · Abdeslam Dennai

Received: date / Accepted: date

Abstract Several features on the face can be identified, like eyes, nose, mouth, lips, etc. These features can be used for many purposes. The most important of which is helping people with disabilities who lose one or all of their senses. This paper presents an algorithm that enables determining the direction of the person's eye in order to manage an intelligent interface where it determines the desired direction. So, that detects the face with the camera, and through the detector MTCNN (Multi-Task Cascaded Convolutional Neural Networks), can be identified the features, and with a map (called landmark points), it is possible to know the coordinates of every feature in the face. We need to define the center(pupil) of the eye, which with his coordinates makes the decision, and the pupil's coordinates calculate through the map and a set of calculations.

Keywords Computer Vision · Face detection · Eye detection · Human-Computer Interaction · image recognition

1 Introduction

Computer vision is among the main areas that have contributed to the development of artificial intelligence and created many challenges. Among these challenges is tracking eye movements. Eye movement is a means by which a person can express his needs and desires. With the help of the eye, the world around us can perceive by taking pictures and sending them to the brain for

S. Benkerzaz
University of Tahri Mohammed Smart Grid & Renewable Energies Laboratory Cloud Comuting & Artificial Intelligence Team Bechar, Algerie
E-mail: benkerzaz.saliha@univ-bechar.dz

Y. Elmir
University of Tahri Mohammed Smart Grid & Renewable Energies Laboratory Cloud Comuting & Artificial Intelligence Team Bechar, Algerie
E-mail: elmir.youssef@yahoo.fr

processing. Checking eye movement contributed to the development of human-computer interaction. This technology permitted the creation of intelligent interfaces that keep in touch with the computer for those who have disabilities, especially mobility or speech disabilities. Eye-tracking applications based on computer vision allow setting the camera on one eye or both eyes. So, two areas in this field are distinguished, the first is to locate the eye's location in the image, and the second to track eye movement and estimate its direction through the data obtained and analyzed. Thus, used directly in applications or tracked via frames in the video in real-time [1]. In this paper, we propose an algorithm that allows detecting the eye and its movement. The second section presents some basic concepts on which the proposed work-based. This section also includes a detailed description of the proposed algorithm. The third section shows the result and discussion. And we conclude at the end of the paper with a conclusion.

2 Proposed Method and Design

The proposed method uses MTCNN to detect the eye area, and a map of landmark points helps determine the direction of the gaze. The platform used here is Python which is a modern computer programming language. It is easy to use and learn and can handle big data that has a mathematical background [2].

2.1 Detector MTCNN

MTCNN is a perfect face detection method proposed by [3] in 2016. This method uses unified cascaded CNNs by multi-task learning. The proposed CNNs consist of three stages:
Stage 1: called Proposal Network (P-Net) that works to provide the candidate windows and their bounding box regression vectors in the same way of [5]. Then it calibrates the candidates by using bounding box regression vectors. Finally, it integrates highly overlapped candidates by non-maximum suppression (NMS).
Stage 2: At this stage, all the candidates are fed to another network called Refine Network (R-Net), where at their level are rejected all the false candidates.
Stage 3: In this stage allows us to focus on more details of the face. As it outputs five facial landmarks' positions. This network is called the Output Network (O-Net).

Although this method can process 7-9 frames per second for 640×360 resolution at full speed [6] it has shown superiority over the latest facial detection methods across several challenging benchmarks like FDDB (Face Detection Data set and Benchmark) [4] and WIDER FACE benchmarks for face detection, and AFLW (Annotated Facial Landmarks in the Wild) benchmark for face alignment [3].

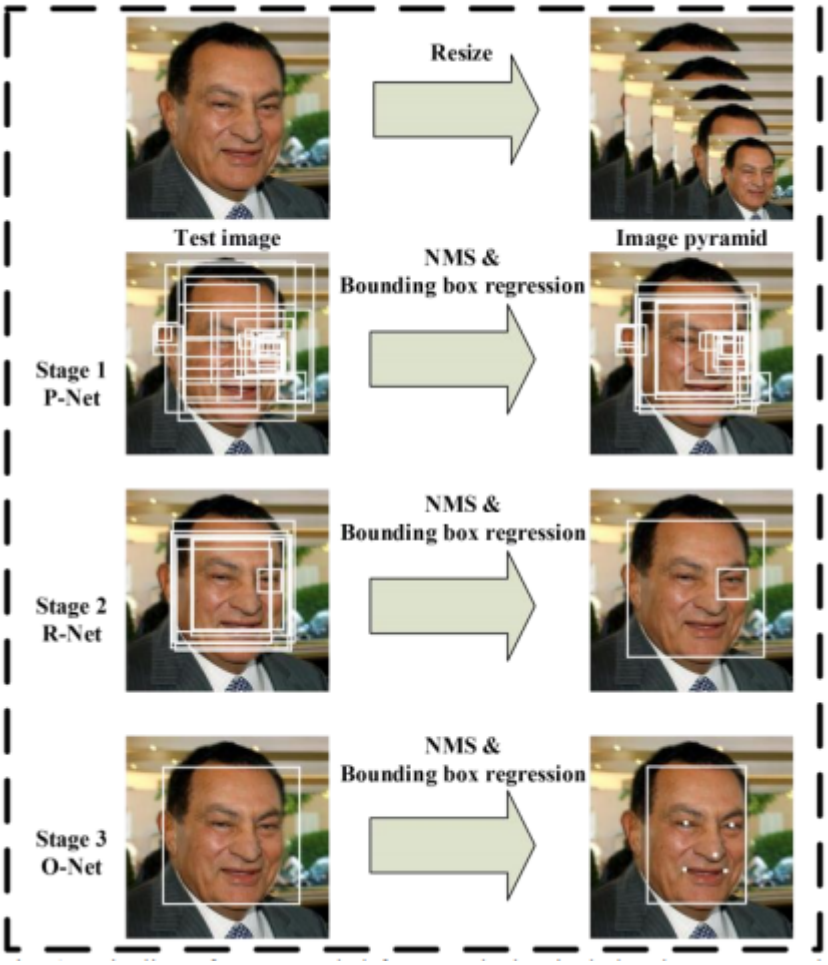

Fig. 1 MTCNN work stages [3].

2.2 Facial landmark detection

Facial landmark detection is an addition to the field of computer vision as it has used in many applications such as face analysis, face recognition, or face modeling [7].
Facial landmark detection algorithms automatically recognize facial key landmark points such as the nose, eyes, and mouth, or dominant points such as the eye corner or lips corners—8]. such each location of point D site has two coordinates x, y. landmark D = Dx; Dy The main objective of detecting facial landmarks is detecting the facial structure using prediction methods, where the detection done in two steps:

– Step 1: identify the face in the image using several methods such as OpenCV's built-in Haar cascades or pre-trained HOG + Linear SVM object detector [9] and can also be used deep learning-based algorithms such as MTCN.

– Step 2: allow the detection of key facial structures in the face region. There are a variety of facial landmark detectors, and all try at localizing and label the following facial regions: Mouth, Right eyebrow, left eyebrow, Right eye, left eye, Nose, Jaw.

The facial landmark detector included in the dlib library uses a manually- labeled training set to specify the (x, y)-coordinates of the regions surrounding each facial structure. And an ensemble of regression trees trained to estimate the facial landmark positions directly. So, the result is a real-time facial landmark detector with high-quality predictions [10].
Dlib is a modern C++ toolkit that contains machine learning algorithms and tools for creating complex programs in C++ and python. It can use in many areas such as robotics, embedded devices, mobile phones, and large high-performance computing environment [11]. The pre-trained face landmark detector inside the dlib library estimates the location of 68 (x, y)-coordinates so that this map coordinated on the facial structures shown in Figure1.[12]

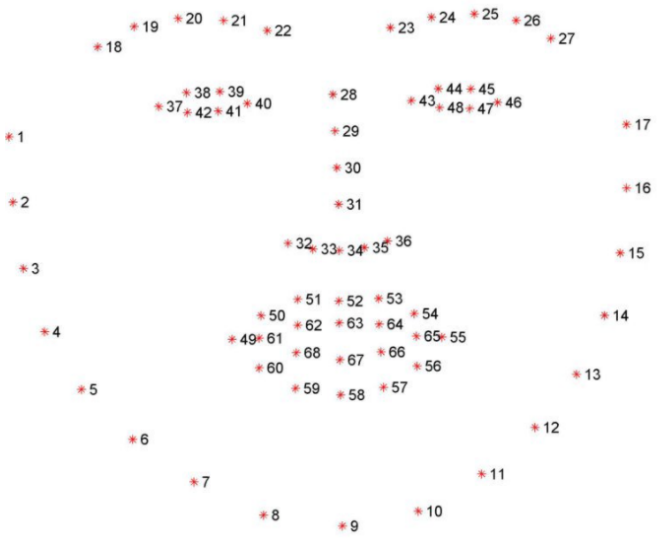

Fig. 2 Visualizing the 68 facial landmark coordinates from the iBUG 300-W dataset [12]

2.3 Proposed Method

The proposed algorithm allows determining the eye direction either to the right or to the left. This algorithm uses the MTCNN detector and Facial landmark map to determine the direction of the pupil of the eye. A facial landmark map can only define the visible landmarks of the face, such as the pupils of the eyes, the corners of the mouth, and the nose. The decision can use to run smart interfaces that intend for either ordinary people or people with disabilities. The algorithm goes through several stages:

- Start by downloading the image,
- Creating a detector MTCNN,
- Detect the face in the image by drawing a box on the face,
- Detect the key points: the pupil, the corners of the mouth, and the nose.
- Calculate the distance Euclidean A, B such as A = distance Euclidean (left eye, nose) B = distance Euclidean (right eye, nose)
- Make a decision.

The following figure shows the step of the algorithm:

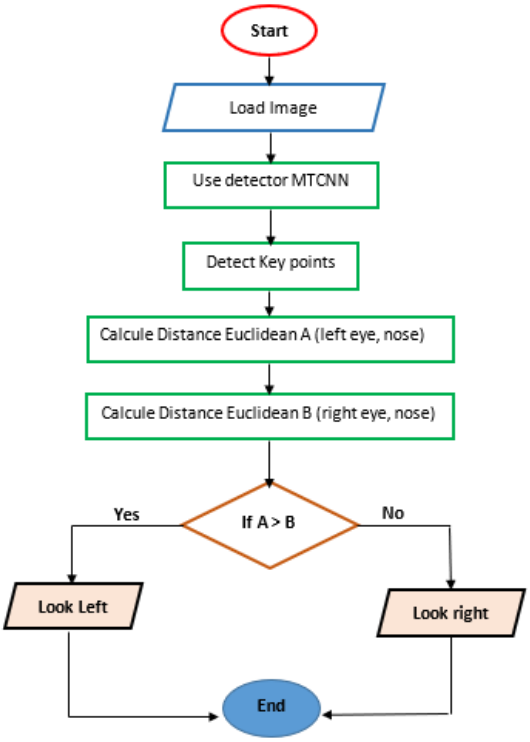

Fig. 3 Algorithm stages .

The algorithm loads the images and then identifies the facial landmark with the MTCNN detector. After that, calculates the Euclidean Distance between the pupil and the nose through the pupil and the nose coordinates. The algorithm decides after comparing the first distance A (for the right eye) and the second distance B (for the left eye). The result defines the face and its landmark coordinates in the image and determines the destination's decision of the eyes' pupil.
We used the Python language, which includes serval libraries of computer vision, the most important of which are dlib, OpenCV. Figure 4 shows the outputs of some images that the algorithm tested on.

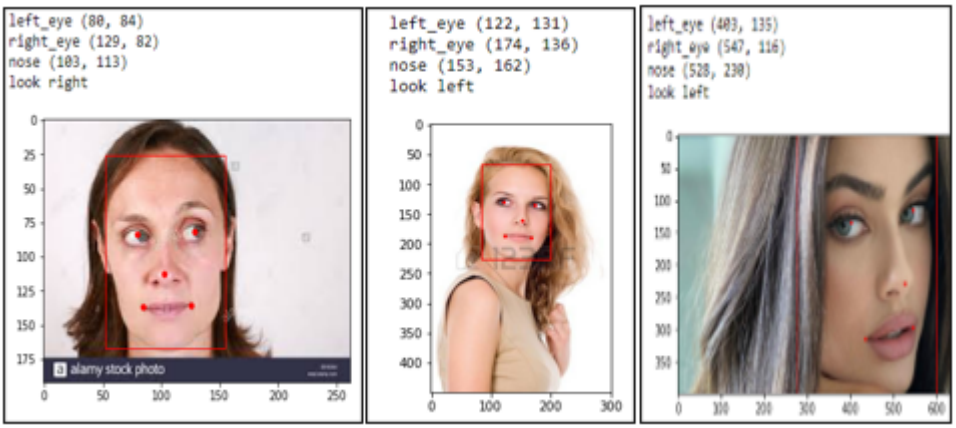

Fig. 4 Stages of detecting landmark coordinates and making a decision.

3 Conclusion

This paper presented an algorithm that allows determining the eye's orientation by calculating the Euclidean distance between the pupil and the nose. The result of this algorithm can use to create smart interfaces oriented to people with disabilities. There are several shortcomings, and we will remedy them in our future work, as we will work on detecting the direction in real-time for many people also at one time.

References

1. Amer Al-Rahayfeh and Miad Faezipour, Eye Tracking and Head Movement Detection: A State-of-Art Survey, IEEE J Transl Eng Health, 1: 2100212 (2013)
2. Afeefa Muhammed, Ramsi Mol, L. Revathy Vijay, S. S. Ajith, A. R. Shamna, Facial Expression Recognition using Support Vector Machine (SVM) and Convolutional Neural

Network (CNN),International Journal of Research in Engineering, Science and Management ,3, Issue-8, (2020)
3. Zhang, K., Zhang, Z., Li, Z., et al.,Joint face detection and alignment using multitask cascaded convolutional networks, IEEE Signal Process, 23,10, 1499–1503,(2016)
4. Jain, V., Learned-Miller,FDDB: a benchmark for face detection in unconstrained settings,Technical Report UM-CS-2010–009, University of Massachusetts, Amherst, (2010).
5. S. S. Farfade, M. J. Saberian, and L. J. Li, Multi-view face detection using deep convolutional neural networks, in ACM on International Conference on Multimedia Retrieval, 643-650,(2015)
6. Long-Hua Ma , Hang-Yu Fan, Zhe-Ming Lu , Dong Tian,Acceleration of multi-task cascaded convolutional networks, IET Image Processing, doi: 10.1049/iet-ipr.2019.0141 www.ietdl.org,(2020).
7. B. Martinez and M. F. Valstar, Advances challenges and opportunities in automatic facial expression recognition, In Advances in Face Detection and Facial Image Analysis, 63–100. Springer,(2016).
8. Yue Wu · Qiang Ji , Facial Landmark Detection: a Literature Survey,International Journal on Computer Vision,doi.org/10.1007/s11263-018-1097-z,(2018).
9. https://www.pyimagesearch.com/2017/04/03/facial-landmarks-dlib-opencv-python/ ,30/06/2021 ,11.44.
10. V. Kazemi, J. Sullivan,One millisecond face alignment with an ensemble of regression trees , IEEE Conference on Computer Vision and Pattern Recognition,(2014)
11. http://dlib.net , 30/06/2021, 19:11.
12. https://towardsdatascience.com/facial-mapping-landmarks-with-dlib-python-160abcf7d672 , 23/06/2021, 15:00.