Docol©c-Report

Digital signed

Docol©c

Reviewed document: **jmcms-2009021 - Face liveness detection (Barnala SARITHA) 9-7-2020.docx**
Processing date: **29.7.2020 9:05 CEST**

A total of 66 sentences were analysed. As a result **19** sentences (28.8%) were found in other documents.

These sentences are highlighted in the text by using different color shades according to the amount of similarity. The darker the highlighting color, the more words were found in another document. You may click the highlighted sentences in order to get further details about found reference documents. Learn more about this report and how to evaluate it.

The following graphic shows the distribution of found sentences within the checked document. The colored parts of the overview bar indicate those parts of the document in which sentences were found in other documents. The left boundary of the bar corresponds to the beginning of the document and the right boundary to the end of the document accordingly. By clicking into the overview bar you are directed to the corresponding position in the document.

Reference documents

The following list contains titles and addresses of documents in which similar sentences were found. With a click on the number of found sentences („**x** Sentences") the corresponding sentences are highlighted in the document as well as in the navigation bar by a colored border and you are directed to the first position of the corresponding sentences in the document. Another click on „**x** Sentences" resets the highlighting.

10 Sentences were found in a text with the title: „***Real-time face liveness detection with Python, Keras and ...***", located at:
https://towardsdatascience.com/real-time-face-liveness-detection-with-python-keras-and-opencv-c35dc70dafd3
https://school.geekwall.in/p/b816a3Pl/real-time-face-liveness-detection-with-python-keras-and-opencv
https://morioh.com/p/25750d792217

6 Sentences were found in a text with the title: „***International Journal of Engineering & Advanced Technology (IJEAT)***", located at:
https://www.ijeat.org/wp-content/uploads/papers/v9i1/A1314109119.pdf

6 Sentences were found in a text with the title: „***IJEAT-Cover - Souvenir_IJEAT_Volume-9_Issue-1_October_2019.pdf***", located at:
https://www.ijeat.org/wp-content/uploads/Souvenir_IJEAT_Volume-9_Issue-1_October_2019.pdf

3 Sentences were found in a text with the title: „***MergedFile - Scopus & SCI List.pdf***", located at:
http://www.griet.ac.in/images/Scopus &amp; SCI List.pdf

2 Sentences were found in a text with the title: „***基于Python+Keras+OpenCV实现实时人脸活体检测 | 文末送书_深 ...***", located at:
https://blog.csdn.net/woshicver/article/details/106700790

▶ In 55 further documents exactly one sentence was found. (click to toggle view)

Subsequent the examined text extract:

Face liveness detection in Real-time using Keras and Open CV

[1]Banala Saritha, [2]Dr.Tejaswini Kar, [3]Dr.Priyadarshi Kanungo

[1]Assistant Professor, S R Engineering College

[2] Assistant Professor, Kalinga Institute of Industrial Technology

[3] Professor, C.V.Raman Global University

[1]banalasarita@gmail.com, [2]tkarfet@kiit.ac.in, [3]pkanungo@gmail.com

Abstract

Face recognition includes both face identification and face verification .Most facial recognition algorithms work well in detecting and recognizing faces on images, videos and video streams from webcam. However these algorithms work on 2-Dimensional frames, but these are not able to discriminate between live faces and photo faces. For facial recognition door opener, it is easy to make a distinction between known faces and unknown faces. But when an ill-intentioned person entered by showing an authorized person's photo it is difficult to detect. The purpose of the paper is to implement face liveness detection algorithm based on eye-blink detection to prevent photo attacks. The algorithm implemented on keras and Open CV with python in windows platform. When a live face is taken through a webcam it displays the person's name when they blinked.

Keywords Open CV, keras, eye-blink detection, face liveness detection algorithm.

1. Introduction

Face detection and recognition can be done in many ways [2,3,4,10]. But it is easy to use **face_recognition** library that provides deep learning methods to detect and recognize the faces. Along with this library the **face_locations**, **face_encodings** and **compare_faces** functions are plays a vital role.

The **face_locations** method can be detect faces using the following methods

1. **Histrogram of oriented Gradients (HoG)**
2. **Convolutional Neural Network (CNN)**.

To encode the given image into features of 128 vectors a Convolutional Neural Network which is a pre-trained network known as **face_encodings** is used. This embedding vector provides enough information to differentiate between two different faces. Two embedding vectors distance can be computed using the function **compare_faces**. *Initially face* is extracted from a webcam frame and then it is compared with the embedding vector obtained from encoded faces from dataset. The closest vectors have minimum distance that corresponds to the same person [5]**.**

Flowchart of the algorithm used:

2. Steps involved in Real-time Face liveness detection using Keras

A. Encoding dataset of known faces

In this paper, the algorithm is designed to recognize two faces myself and Trisha Krishnan. I have taken nearly 50 pictures of each. The following code is used to encode our database of known faces.

```
encoding = face_recognition.face_encodings(image, boxes)
```

Next, we need to differentiate between a photo face and a live face.

B. Real time face liveness detection

To classify whether an eye is closed or open, a trained a Convolutional Neural Network is required. The LeNet-5 CNN model was chosen that has been trained on the Closed Eyes in the Wild (CEW) dataset. It consists of approximately 4800 eye images in size of 24x24.I got 94.4% accuracy for 20 epochs when the model is evaluated. The following libraries are required to train the CNN by using the following code.

```
from keras.models import Sequential

from keras.layers import Conv2D

from keras.layers import AveragePooling2D

from keras.layers import Flattern

from keras.layers import Dense

from keras.preprocessing.image import ImageDataGenerator
```

Whenever an eye is detected [1] its status can be predicted using this model, and the process will be continued for each person to keep track of the eyes status [8,9].

Therefore, it becomes really easy to detect an eye blinking [6, 7] using the following function .It tries to find a closed-open-closed pattern in the eyes status history.

Eye blinking function: def isBlinking(history, maxFrames):

```
        for i in range(maxFrames):

        pattern = '1' + '0'*(i+1) + '1'
```

```
if pattern in history:

return True

return False
```

Fig1: closed eye dataset given for training

Fig2: opened eye dataset given for training

C. Face recognition of living people

A Real-time Face recognition algorithm can be performed once the known faces encoding dataset (Closed Eyes In The Wild (CEW) dataset) and trained CNN (LeNet-5) are ready. Now by using OpenCV with pre-trained Haar-cascade classifier we can detect faces and eyes in real-time using the following code.

```
def detect_and_display(model, video_capture,face_detector,open_eyes_detector, left_eye_detector, right_eye_detector, data, eyes_detected):

frame = video_capture.read()

# resize the frame

frame = cv2.resize(frame, (0, 0), fx=0.6, fy=0.6)

gray = cv2.cvtColor(frame, cv2.COLOR_BGR2GRAY)

rgb = cv2.cvtColor(frame, cv2.COLOR_BGR2RGB)

# Detect faces

faces = face_detector.detectMultiScale(gray, scaleFactor=1.2,minNeighbors=5,

minSize=(50, 50),

flags=cv2.CASCADE_SCALE_IMAGE)
```

The function def detect_and_display uses the arguments model, video_capture, face_detector, open_eyes_detector, left_eye_detector, right_eye_detector, data, eyes_detected. The functionality of arguments are mentioned below.

Argument	Purpose
model	open/closed eyes classifier
video_ capture	To get stream video
face_ detector	Haar-cascade face classifier ***haarcascade_frontalface_alt.xml***
open_ eyes_ detector	Haar-cascade open eye classifier ***haarcascade_eye_tree_eyeglasses.xml***
left_ eye_ detector	Haar-cascade left eye classifier used to detect open or closed eyes.***haarcascade_lefteye_2splits.xml***
right _eye_ detector	Haar-cascade right eye classifier used to detect open or closed eyes. ***haarcascade_righteye_2splits.xml***
data	Set of known encodings and known names

eyes_ detected	Set of eyes status history containing for each name

3. Results:

An effective algorithm for the face liveness detection of human being in real time based on eye blinking using keras and OpenCV is performed. This algorithm declares the liveness of a face i.e whether the face is photo face or live face using the blinking function. The libraries `OpenCV 4.2, Keras2.2.4` a deep learning framework based on the Tensor flow and few supporting libraries are `CV2, numpy,tqdm,imutils, PIL` are used. The environments used are executed on Windows 10 Operating System with python3.7.7.

Fig 3: Person placed in front of camera Fig 4: Person from photo placed in front of camera

Fig 5: Real time human and photo facing camera for liveness detection.

4. CONCLUSION

Initially from the webcam stream a frame is grabbed and resized to speed up computations. Then faces are detected from the frame and encoded into a 128-d vector. These vectors are compared with the known face encodings so that eyes are detected into face boxes. With the help of function open_eye_detector first detected the eyes whether it is open or not? If the detector succeeds, it gives the conclusion that the eyes are open. Since the open_eye_detector cannot detect closed eyes. The left_eye and right_eye detectors are used if the first classifier has failed, then the face is separated into left and right side for the respective detectors to be classified. Then eye part is extracted and the trained model predicts whether the eyes are closed. If one closed eye is detected, then both eyes are predicted to be closed and a '0' is added to the eyes status history. Otherwise it's concluded that the eyes are open. Finally the isBlinking() function is used to detect eye blinking and the person has blinked eye it concluded a liveness face of a real time image. This work produces appreciable results with 94.4% accuracy for 20 epochs when the model is evaluated.

REFERENCES

1. T Soukupova and Jan Cech. Real-time eye blink detection using facial landmarks. In 21st Computer Vision Winter Workshop (CVWW 2016), pages 1–8, 2016.

2. Exploration of active pixels in intelligent face recognition for constrained mobile systems Proceedings of the 5th Indian International Conference on Artificial Intelligence, IICAI 2011, 2011 | conference-paper EID: 2-s2.0-84872182449 Source: Scopus – Elsevier.

3. Methodological approach for machine based expression and gender classification2009 ,IEEE International Advance Computing Conference, IACC 2009, 2009 | conference-paper, DOI: 10.1109/IADCC.2009.4809216, EID: 2-s2.0-66249128289,Source: Scopus - Elsevier.

4. An investigation into the use of partial face in the mobile environment Lecture Notes in Computer Science (including subseries Lecture Notes in Artificial Intelligence and Lecture Notes in Bioinformatics)2011 | book DOI: 10.1007/978-3-642-24031-7_53EID: 2-s2.0-80053355971 Source: Scopus – Elsevier

5. Face Liveness Detection Using a Flash against 2D Spoofing Attack, Patrick P. K. Chan, Member, IEEE, Weiwen Liu, Danni Chen, Daniel S. Yeung, Fellow, IEEE, Fei Zhang, Xizhao Wang, Fellow, IEEE, Chien-Chang Hsu, Member, IEEE, IEEE TRANSACTIONS ON INFORMATION FORENSICS AND SECURITY, VOL. XX, NO. X, OCTOBER 2016

6. Anna Rita Bentivoglio, Susan B Bressman, Emanuele Cassetta, Donatella Carretta, Pietro Tonali, and Alberto Albanese. Analysis of blink rate patterns in normal subjects. Movement Disorders, 12(6):1028–1034, 1997.

7. T. Danisman, I. Bilasco, C. Djeraba, and N. Ihaddadene. Drowsy driver detection system using eye blink patterns. In Machine and Web Intelligence (ICMWI), Oct 2010.

8. C.Anuradha, R.Kavitha,Liveness Detection with Opencv International Journal of Innovative Technology and Exploring Engineering (IJITEE) ISSN: 2278-3075, Volume-8, Issue-9S3, July 2019

9. M. Divjak and H. Bischof. Eye blink based fatigue detection for prevention of computer vision syndrome. In IAPR Conference on Machine Vision Applications, 2009.

10. F. Abate, M. Nappi, D. Riccio, and G. Sabatino, “2d and 3d face recognition: A survey,” Pattern Recognit. Lett., vol. 28, no. 14, pp. 1885– 1906, Oct. 2007.