Title Facial Liveness Testing for the Web - an analysis of different methods

Project Type Computer Vision, Image Processing and Security

Description In order to avoid spoofing in facial recognition systems, liveness tests are needed. While various liveness tests exist, some require specialized hardware and are therefore not suitable. This project aims to select the most suitable methods for a web facial authentication system, and analyze their effectiveness in terms of security. A liveness test is suitable if it uses only one built-in camera (found within laptops in the webcam, or in mobile devices as the front camera), and potentially the device screen (which varies in size depending on the device used), and can be done in near real time.

Preliminary Preparation
- Existing liveness tests
- What are the datasets available for testing/training of these liveness methods?

Minimum Objectives
- Build the test framework, to test using several different datasets and liveness tests.
- Implement the image quality based liveness test.
- Train the classifier used within the image quality based liveness test.
- Evaluate the image quality liveness test.

Intermediate Objectives
- Implement the eye tracking liveness test.
- Train the eye tracking liveness test classifier.
- Evaluate the eye tracking liveness test.
- Implement the CNN based liveness method (involving texture and temporal metrics).
- Train the CNN based liveness method classifier.
- Evaluate the CNN based liveness method.
- Compare CNN, eye tracking, and image quality based liveness methods together.

Advanced Objectives
- Implement a consolodation layer, combining the metrics above through a classifier.
- Train the consolodation layer classifier
- Evaluate the performance of the consolodation layer (i.e. all methods when integrated together) compared to each individual method on it's own.
- Implement the facial flashing liveness test.

References
- Keras (`https://keras.io`) for Machine Learning
- OpenCV (`https://opencv.org/`) for Image Processing
- Image quality based liveness test (`https://ieeexplore.ieee.org/document/6671991`)
- Eye tracking liveness test (`https://waset.org/publications/5308/liveness-detection-for-embedded-face-recognition-system`)
- CNN based liveness test (`https://arxiv.org/pdf/1408.5601.pdf`)
- Facial Flashing liveness test (`https://arxiv.org/pdf/1801.01949.pdf`)

Facial Liveness Tests

Creator	Ryan Collins
DATE	10/18/18

TASK TITLE	Durham Week	14	15	16	17	18	19	20	21	22	23	24	25	26	27	28	29	30	31	32	33	34	35	36	37	38	39	40	41
Dataset Integration																													
NUAA dataset integration	2																												
REPLAY-ATTACK dataset integration	2																												
Test Framework																													
Generic Model Implementation	1																												
Dataset Manager (for test data) using datasets above	1																												
Test Runner	2																												
Quality Liveness Test																													
ReferenceImage creator (Gaussian subtraction)	1																												
Implementation of simple metrics (1-19)	3																												
Implementation of specialised metrics (20 onwards)	3																												
Vector consolodation (providing the vector from the metrics)	1																												
Linear classifier layer	3																												
Eye Tracking Liveness Test																													
Eye Detection	1																												
Face Region Normalization	2																												
Eye region binarization	1																												
Variation calculator	1																												
Variation classifier	1																												
CNN Liveness Test																													
Facial localization	2																												
Spatial augmentation	2																												
Temporal augmentation	2																												
Convolutional Neural Network (ImageNet)	2																												
Consolodation Layer																													
Vector creator based on the output of other tests	1																												
Training a classifier to give an output based on input	1																												
Comparison Data Gathering																													
Comparison of each three models individually	1																												
Models seperately, or models with the consolodation layer	1																												
Face Flashing Liveness Test																													
Generation of parameters	1																												
Initialization of front end	1																												
Presentation of challenge	1																												
Response collection	2																												
Repetition of response	1																												
Timing Verification	3																												
Face verification	3																												
Expression verification	2																												