**Image splicing detection**

L’algoritmo, presentato in [1], consente la classificazione di un’immagine contenente almeno due facce come autentica o falsificata. In particolare è possibile andare ad individuare, nel caso di un’immagine falsa, quale delle facce presenti nell’immagine è stata manomessa.

**Algoritmo**

Viene descritta a seguito la procedura implementata per il **training**.

1. **Illuminant maps extraction**

Per ciascuna immagine del dataset, estrae la GGE e la IIC map dell’intera immagine. Le due mappe vengono considerate come un nuovo spazio colore dell’immagine originale. L’estrazione delle mappe viene effettuata con la tecnica descritta da Riess in [2].

1. **Face detection and paired face feature extraction**

Eseguita l’estrazione delle due mappe, viene letto dal groudtruth la posizione delle facce.

Se l’immagine contiene un numero maggiore di due facce si continua con il processo, altrimenti l’immagine viene scartata.

Per ciascuna faccia presente, si ricavano le due mappe (GGE e IIC) relative a quest’ultima (selezionando la regione nelle stesse corrispondente) e si esegue un algoritmo di estrazione di un descrittore di colore. I descrittori presi in considerazione sono il ACC, BIC, CCV e LCH.

A questo punto per ciascuna coppia di facce presenti ne viene creato un descrittore concatenandone i due descrittori: una coppia sarà considerata come falsificata se almeno una delle due facce presenti risulta essere falsa, altrimenti sarà considerata come autentica. (N.B.: allo stato attuale non si tiene conto dell’ordine delle due facce, ma in realtà il descrittore cambia in base all’ordine. E’ una cosa da provare)

1. **KNN models training**

Per la fase di training viene innanzitutto selezionato un valore di K per l’addestramento dei modelli KNN.

Per ciascun diverso descrittore viene poi creato un modello (i.e. un modello che utilizza descrittori AAC, uno per BIC, etc.).

(N.B.: allo stato attuale viene considerato solamente descrittori ottenuti processando le GGE map. Da estendere anche all’utilizzo di descrittori su IIC map).

Viene descritta a seguito la procedura implementata per la **detection**.

1. **Illuminant maps extraction**
2. **Face detection and paired face feature extraction**

Come nel caso precedente, ma la detection delle facce viene effettuata tramite l’algoritmo di Viola e Jones, con la sua implementazione per OpenCV.

1. **KNN classification**

Ciascuna coppia di facce viene ora classificata tramite i modelli KNN addestrati. Per ciascun classificatore otteniamo una prediction, il risultato è ottenuto tramite votazione a maggioranza tra tutti i classificatori: una coppia di facce sarà considerata come falsa se almeno la metà più 1 dei classificatori utilizzati la classifica come non autentica.

1. **Single face choose**

Data una coppia di facce considerata falsa, si procede alla scelta della faccia falsa tra le due tramite il metodo descritto anche in [3]. L’intuizione nell’articolo di Victor e Tiago che, data un’immagine vera ed una contenente una contraffazione, la differenza tra le due Illuminant Maps estratte (GGE e IIC) è maggiore in quella contraffatta. Con questa intuizione viene stabilito che la faccia falsificata è quella che presenta una maggiore differenza tra le due mappe colore.

*N.B. : questa intuizione purtroppo però non si è dimostrata solida, portandoci a cambiare direzione verso questo altro approccio.*

**Dataset**

I dataset utilizzati per i test del procedimento sono il DSO-1 e il DSI-1, entrambi reperibili in rete e creati da Tiago.

<https://recodbr.wordpress.com/code-n-data/#dso1_dsi1>

DSO-1 It is composed of 200 indoor and outdoor images with an image resolution of 2,048 x 1,536 pixels. Out of this set of images, 100 are original, i.e., have no adjustments whatsoever, and 100 are forged. The forgeries were created by adding one or more individuals in a source image that already contained one or more persons.

DSI-1 It is composed of 50 images (25 original and 25 doctored) downloaded from different websites in the Internet with different resolutions. Original images were downloaded from Flickr and doctored images were collected from different websites such as Worth 1000, Benetton Group 2011, Planet Hiltron, etc.

Il groundtruth è composto da un file di testo per ciascuna immagine così popolato:

*1 FAKE 1112 1502 496 952*

*2 NORMAL 680 1100 388 890*

*…*

In ordine di apparizione:

* *Il numero della faccia nell’immagine (progressivo)*
* *La classificazione (FAKE o NORMAL)*
* *Top left x*
* *Top left y*
* *Bottom right x*
* *Bottom right y*

**Esperimenti**

1. **Classificazione con crossvalidazione coppie di facce (ACC) (10/01/2017)**

Training e valutazione delle performance con crossvalidazione (10-fold) del DSO-1 utilizzando un solo classificatore addestrato con descrittori ACC.

Accuracy (avg over 5 runs): 70.2%

1. **Classificazione con crossvalidazione coppie di facce (ACC e BIC) (11/01/2017)**

Training e valutazione delle performance con crossvalidazione (10-fold) del DSO-1 utilizzando un solo classificatore addestrato con descrittori ACC e BIC.

Accuracy (avg over 5 runs): 72.2%

1. **Classificazione con crossvalidazione coppie di facce (ACC, BIC, CCV) (11/01/2017)**

Training e valutazione delle performance con crossvalidazione (10-fold) del DSO-1 utilizzando un solo classificatore addestrato con descrittori ACC, BIC e CCV.

Accuracy (avg over 5 runs): 73.2%

1. **Classificazione con crossvalidazione coppie di facce (ACC, BIC, CCV e LCH) (16/01/2017)**

Training e valutazione delle performance con crossvalidazione (10-fold) del DSO-1 utilizzando un solo classificatore addestrato con descrittori ACC, BIC, CCV e LCH.

Accuracy (avg over 5 runs): 75.6%

**Codice sorgente**

Linguaggio utilizzato Python, versione 3.4.

Librerie utilizzate: OpenCV 3.0

Tutto il codice sorgente è completamente aperto e disponibile su <https://github.com/lorecioni/ImageSplicingDetection>.

*Generazione delle Illuminant Maps*

Linguaggio C++

Dipendenze:

- OpenCV (tested with 2.4.0)

- Boost (tested with 1.45)

- cmake (tested with 2.8.2)

<https://www5.cs.fau.de/en/research/areas/computer-vision/image-forensics/scene-illumination-as-an-indicator-of-image-manipulation/>

*Estrazione dei descrittori ACC, BIC, CCV, EOAC, LAS, LCH, SASI, SPYTEC, UNSER*

Linguaggio C++

**Funzione principale: main.py**

ImageSplicingDetection v.0.2

Creation date: 2016-09-12, last update: 2017-01-17

usage: main.py [-h] [--train] [--detect] [--cross-validate]

[--extract-single-features] [--no-extract-features]

[--no-extract-maps] [--img IMG] [--heat-map] [--verbose]

optional arguments:

-h, --help show this help message and exit

--train train the model for further splicing detection

--detect detect splice over an image

--cross-validate cross-validate the dataset

--extract-single-features extract feature vector for a specific image

--no-extract-features no extract all training images features

--no-extract-maps no extract all training images features

--img IMG the path of the suspicious image

--heat-map display the heat map between GGE and IIC maps

--verbose display all messages

References:

[1] T. Carvalho, et al. *Illuminant-Based Transformed Spaces for Image Forensics*. IEEE Transactions on Information Forensics and Security 11.4 (2016): 720-733.

[2] C. Riess and E. Angelopoulou, *Scene illumination as an indicator of image manipulation*. In Proceedings of the 12th international conference on Information hiding, Berlin, Heidelberg, 66-80; 2010

[3] V. Schetinger et al. *Exploring Statistical Differences Between Illuminant Estimation Methods for Exposing Digital Forgeries*; 2016.