Illuminant-based Transformed Spaces for Image Forensics

Tiago Carvalho, Fábio A. Faria, Hélio Pedrini, Ricardo da S. Torres, and Anderson Rocha.

In IEEE Transactions on Information Forensics and Security, v.0, no.00, p. 0-0, 2015.

1 Overview

This material presents details regarding source code and datasets used in experiments performed at the paper *Illuminant-based Transformed Spaces for Image Forensics*. All datasets are available in `http://www.ic.unicamp.br/~rocha/pub/DB_IBTSFIF.tar.gz`. Source code of illuminant maps generation, image descriptors and machine learning framework used in features selection are available in `http://www.ic.unicamp.br/~rocha/pub/SC_IBTSFIF.tar.gz`.

2 Datasets description

When dealing with image splicing forgeries containing people (more than one person), the images under analysis are doctored in a way that the forgery looks as perfect as possible. However, this images also can pass through many image processing operations (e.g. compression, resize, bright adjustments). To show that our approach is not biased on how the images were created, the dataset of experimental evaluation comprises three scenarios: one with images in high quality and very convincing forgeries (*DSO-1*), a second one with images downloaded from different sources at the Internet (and susceptible to different image processing operations) with different resolutions (*DSI-1*), and another one with a few questioned images downloaded from Internet involving famous people (*FMS-1*) [1].

2.1 DSO-1

The dataset was created by Carvalho *et.al.* [1]. It is composed of 200 indoor and outdoor images with an image resolution of $2,048 \times 1,536$ 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. When necessary, we complemented an image splicing operation with post-processing operations (such as color and brightness adjustments) in order to increase photorealism.

2.2 DSI-1

Also created by Carvalho *et.al.* [1], it is composed of 50 images (25 original and 25 doctored) downloaded from different websites in the Internet with different resolutions.

2.3 FMS-1

The FMS-1 dataset comprises five questioned images which have been broadcasted by internet. These images were analyzed qualitatively and are helpful to show robustness of the proposed method.

[1] In addition to all datasets, we are also providing all the illuminant maps (calculated from IIC and GGE method) and the faces positions and labels for each image inside the dataset. Also, we included the 5-folds used in cross-validation protocol

3 Source Code

The source code of proposed methods have three main parts:

- Illuminants estimation using IIC and GGE;
- Image descriptors extraction;
- Machine learning framework to select the best set of descriptors (available soon);

Each one of aforementioned parts is better described in the next sections.

3.1 Illuminant Estimation

To estimate illuminant maps from images is necessary to follow this steps:

1. enter *./illuminants/*, open the *README.txt* file and follow the instructions in the file to compile the *Vole* module;
2. copy the file *./illuminants/config.txt* and paste it inside the folder *../illuminants/build/bin/* ;
3. enter *./database/images/* and insert your images into this folder (if there are any other file, delete them before);
4. execute the function *./source-code/segmentAllImagesForIlluminantMethod.py* to segment all the images and prepare the illuminant maps;
5. to estimates the illuminant maps using IIC method, execute the function *./source-code/extractIICMaps.py*. Resulting maps are placed on *./database/IIC/*;
6. to estimates the illuminant maps using GGE method, execute the function *./source-code/extractGGEMaps.py*. Resulting maps are placed on *./database/GGE/*;

3.2 Image Descriptor Extraction

Once illuminant maps have been extracted, to extract the image descriptor from images is necessary to follow this steps:

1. for each descriptor inside *./descriptors/* do (below, we will use ACC descriptor as an example):
   (a) enter *./descriptors/acc/source/* and execute *make clean* command;
   (b) also in *./descriptors/acc/source/* execute *make* command;
   (c) enter *./descriptors/acc/source/app/* and execute *make acc_extraction* command (without .c extension);
2. Before to be continues point the steps to extract image descriptors from your illuminant maps, it is necessary to highlight that, for each image in your data, you need to provide a *.txt* file with the same name of the image, which will be placed on *./face-positions/*. This file needs to contains the face positions (a bounding box around the face delimited by top left and bottom right points) of all faces in your image. Each line of this file *.txt* contains information about one face at the image in the following structure (insert tab between columns):

   id *facelabel* *xtopleft* *xbottomright* *ytopleft* *ybottomright*

   One example of image position file is placed on *./face-positions/position.txt*. This file would be associated with an image named *position* containing two faces: the first one is a NORMAL face while the second one is a FAKE face. Make sure that, when you are creating a position file for your image, we use the labels NORMAL and FAKE to point pristine and fake faces, respectively.
3. Make sure Python 3.4 iand library *OpenCV* (cv2) are installed

4. To extract image descriptors from illuminant maps (IIC and GGE) of all of the images into *./database/images/* folder, for an specific descriptor, execute the function *./source-code/extractAllFeatureVectors.py <DS>* where:

    - *DS:* an string representing the descriptor to be extracted. Is possible to extract nine kinds of descriptor as described by Carvalho et. al [2] (ACC, BIC, CCV, EOAC, LAS, LCH, SASI, SPYTEC, UNSER). If you want to extract all of the descriptors, replace <DS> by "ACC BIC CCV EOAC LAS LCH SASI SPYTEC UNSER";
    - Resulting files are recorded at *./extracted-feature-vectors/DS-IM-CS-CC/* where IM is the illuminant map used (GGE or IIC), CS is the color space (HSV, RGB, YCbCr, Lab) and CC is the channel used (full indicates that we have been used a color image and not an specific channel).
    - Remember that the folder *./face-positions/* needs to contains one position file for each image as described at Section 3.2 step 2.

References

[1] CARVALHO, Tiago Jos de; RIESS, Christian; ANGELOPOULOU, Elli; PEDRINI, Hélio; ROCHA, Anderson de Rezende. Exposing Digital Image Forgeries by Illumination Color Classification. IEEE Transactions on Information Forensics and Security (T.IFS), v. 8, p. 1182-1194, 2013.

[2] CARVALHO, Tiago José de; FARIA, Fábio; PEDRINI, Hélio; TORRES, Ricardo da Silva; ROCHA, Anderson de Rezende. Illuminant-based Transformed Spaces for Image Forensics. IEEE Transactions on Information Forensics and Security (T.IFS), v. 0, p. 00-00, 2016.