RESEARCH-ARTICLE

Deep Learning-based Facial Detection and Recognition in Still Images for Digital Forensics

PATRÍCIO DOMINGUES, School of Technology and Management, Leiria, Leiria, Portugal

ALEXANDRE FRAZÃO ROSÁRIO, School of Technology and Management, Leiria, Leiria, Portugal

Published: 26 August 2019

ARES '19: 14th International Conference on Availability, Reliability and Security
August 26 - 29, 2019
CA, Canterbury, United Kingdom

Deep Learning-based Facial Detection and Recognition in Still Images for Digital Forensics

Patricio Domingues*
ESTG - Polytechnic Institute of Leiria
Instituto de Telecomunicações
Centro de Investigação em Informática e Comunicações
Leiria, Portugal
patricio.domingues@ipleiria.pt

Alexandre Frazão Rosário*
ESTG - Polytechnic Institute of Leiria
Instituto de Telecomunicações
Leiria, Portugal
alexandrefrazaorosario@gmail.com

ABSTRACT

Smartphones and cheap storage have contributed to a deluge of digital photos. Digital forensic analysis often include the need to process large volumes of digital photos found on devices. Sometimes, this is done either to detect or confirm the ownership of the device or to determine whether the owner of the device has some acquaintance of interest in the case. In this paper, we present the Face Detection and Recognition in Images (FDRI) open source software, and its integration as a module for the digital forensic software Autopsy. FDRI aims to semi-automate the detection of faces in digital photos, flagging photos where at least one face is detected. FDRI software also performs face recognition, searching for the existence of given individual(s) in still photos of the forensically examined devices. For both the detection and recognition of faces, FDRI resorts to deep learning-based algorithms available within the *dlib* machine learning toolkit. In experimental assessments, FDRI yielded an average precision of 99.46% face detection and 98.10% for face recognition, when dealing with the restrained LFW dataset. For unrestrained real world photos, FDRI achieved a precision of 97.67% for face detection and 81.82% for face recognition. Performance-wise, this study confirms the importance of a fast GPU for fast face detection and recognition, with an NVidia GTX 1070 being roughly three times faster than a GTX 750 Ti, and in certain cases, up to 35× faster than the CPU version.

CCS CONCEPTS

• **Applied computing** → **Computer forensics**; **Evidence collection, storage and analysis**.

KEYWORDS

face detection, face recognition, digital forensics, deep learning

ACM Reference Format:
Patricio Domingues and Alexandre Frazão Rosário. 2019. Deep Learning-based Facial Detection and Recognition in Still Images for Digital Forensics. In *Proceedings of the 14th International Conference on Availability, Reliability and Security (ARES 2019) (ARES '19), August 26–29, 2019, Canterbury, United Kingdom.* ACM, New York, NY, USA, 10 pages. https://doi.org/10.1145/3339252.3340107

*Both authors contributed equally to this research.

ARES '19, August 26–29, 2019, Canterbury, United Kingdom

ACM ISBN 978-1-4503-7164-3/19/08…$15.00
https://doi.org/10.1145/3339252.3340107

1 INTRODUCTION

Digital age has revolutionized photography. While analog photography was costly, physically limited and with a significant time delay between photo exposure and the final result, digital has made it cheap, instantaneous and high quality. First confined to digital cameras, the emergence of smartphones with increasing quality cameras have made digital photography even more ubiquitous. Cheap storage and ubiquitous Internet connectivity contributed to make photo storage and sharing easy, further facilitating digital photography. Affordable storage allows for keeping large volume of data, namely digital photos, at a small price. For instance, considering 10 MP JPEG photos, with each one roughly requiring 3 MiB of storage, a 32 GB micro SD card can store approximately 10 000 photos. Beaver et al. report that in 2010, an average of 10^9 photos were uploaded every week to Facebook [3]. Wakabayashi pinpoints that Fuji estimated that 1.6×10^{12} digital photos were taken in 2013 [25]. This growth has certainly continued since then, and it is expectable that the trend continues, at least in the near future.

Digital forensics is facing a device and data deluge crisis [7][20]. According to Quick & Choo, three main causes contribute to the data deluge: *i*) more seized devices; *ii*) more cases requiring digital forensics examination and *iii*) more data on each individual item [20]. All of this increases the demands on digital forensics examinations, leading to more examinations and lengthier ones, and thus increasing backlogs of digital forensic laboratories. The easiness for taking photos and record videos brought by smartphones, further contributes to lengthen digital forensics examinations, when the case requires processing multimedia sources such as photos and videos. Other sources of digital photos, such as cache browsers can further increase the workload for processing digital photos. Indeed, manually analyzing photos and videos is a labor intensive, time consuming and error prone task.

In this work, we resort to the deep learning algorithms of the *dlib* machine learning kit (http://dlib.net) for the detection and recognition of faces in digital photos. The goal is to speed up the process of detecting and recognizing faces in digital photos, presenting digital forensic experts with the photos where faces were detected and, if solicited by the experts, where wanted faces were recognized. Examples of use cases include the need to report whether a given

individual is present in any photos of the case (e.g., a stalking case), to prove that two or more individuals know each other, or at least were simultaneously at a given place, as documented by a photo, or to assert that the suspect has, in the past, been in a given event or place. For this purpose, we have developed the *FDRI* software, short for *Face Detection and Recognition in Images*. Although we present FDRI within the context of the digital forensic tool Autopsy, FDRI can be run as an independent external program, or integrated with another digital forensic tool, as FDRI receives input parameters through a JSON-based file and outputs results using the DFXML format [6]. FDRI has two main modes: *i*) face detection and *ii*) face recognition. The first mode aims to report all digital photos where at least one human faces is detected. This way, instead of combing through thousand of photos, the digital forensic expert can, whenever relevant, examine solely photos where faces have been detected. The second mode provides for face recognition: the digital forensic expert provides at least one photo with the face of the wanted individual, and FDRI analyzes the photos of the digital forensic dataset looking for photos where the sought individual(s) is present.

The main contribution of this work is the ability to semi-automate and speed up face detection and face recognition in digital photos, saving precious time to digital forensic experts, so that they can focus on tasks that truly require human expertise. By semi-automate, we mean that results produced by FDRI need validation by an human expert, as precision of face detection and face recognition in unconstrained real world photos is still not 100% reliable [1, 2]. Our contribution takes the form of an open source Jython-based module for the Autopsy digital forensic software and an associated program – *FDRI.exe* – built on top of the dlib library. Another contribution of the paper is the analysis of the *i*) precision of the proposed solutions and *ii*) measuring the computational performance through execution times. Both metrics are assessed with the well known *i*) *Labeled Faces in the Wild* dataset, with *ii*) a forensic dataset of a daily-used laptop and with *iii*) a set of unconstrained real photos.

The remainder of the paper is organized as follows. Section 2 reviews related work. Section 3 briefly reviews the Autopsy software, while FDRI and its interaction with Autopsy are presented in Section 4. Section 5 describes the experimental setting and analyzes the main precision and computational performance results. Finally, Section 6 concludes the paper and present venues for future work.

2 RELATED WORK

We review related work, first focusing on face detection and then on face recognition. Due to the huge volume of scientific literature on face detection and on face recognition, we mainly focus on face detection and face recognition applied to digital forensics.

2.1 Face Detection

Zafeiriou et al. [28], in their comprehensive survey, organize face recognition algorithms in two major categories: *i*) algorithms based on rigid-templates and *ii*) algorithms that learn and apply a Deformable Parts-based Model (DPM) and which are used for developing generic object detectors [28]. Zafeiriou et al. further classify the rigid-templates into three main family of algorithms: *i*) Viola-Jones [24] face detection algorithm and variations, broadly identified as Haar-cascade; *ii*) methods based on image retrieval and Histograms of Oriented Gradients (HOG) [23] and, finally, *iii*) algorithms based on Convolutional Neural Networks (CNN) and deep CNNs [14].

Due to their lower preparation demands, Haar-cascades and HOG methods are often used for real-time face detection, namely in digital cameras, smartphones and surveillance systems. These methods are behind the familiar square box that appears on the screen of some digital cameras and digital photography applications, or help to detect the best instant for a photo, namely when the subject is smiling. CNN algorithms require a more elaborate setup, namely for training, are more computationally demanding, and thus are not yet appropriate for most mobile devices. However, they yield better detection results. Examples of wide-scale usage include Facebook's face tagging [26] and Google reverse search images.

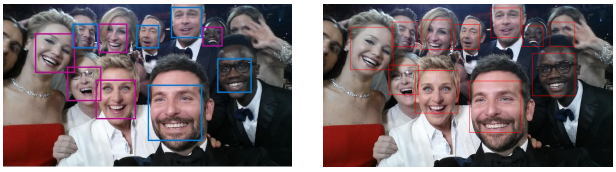
FDRI could resort to Haar-cascades or HOG methods for face detection since there are faster than CNN-based detection methods. Indeed, for digital forensics purpose and for the sole purpose of face detection, it is not of the utmost importance to detect all faces of a photo, since any photo with at least one detected face is flagged to be manually reviewed by the digital forensic expert. However, two situations are determinant for focusing on an high precision face detection methodology: *i*) false negatives, that is, when no face is detected in a photo that has at least one face, and thus the photo is not flagged; *ii*) face detection is the first step of face recognition, since face recognition lookup is only applied to detected faces. Therefore, failing to detect a face also hinders the success of face recognition.

Strothers and Boonthum present the open source module FaceRadar for Autopsy [4]. The module aims to detect faces that exist in images. FaceRadar resorts to OpenCV's implementation of the Haar cascade face detection algorithm [24]. The authors report that tests on several sets of images yielded an average precision of 90.61% and average recall of 98.87% [4]. However, no quantitative data are provided regarding the execution speed. Additionally, despite our best efforts, we could not locate the source code nor the executable of FaceRadar. Our work departs from FaceRadar, since FDRI also provides face recognition. Moreover, FDRI resorts to deep convolutional neural network-based algorithms for face detection/recognition, employing what is currently considered state-of-the-art algorithms. In fact, as shown in Figure 1, dlib/FDRI detects 11 faces in the photo (right), while Microsoft Face API (left) detects ten, missing the top right one. To be fair, it should be pointed out that Microsoft Face API goes far beyond face detection, since it provides estimation for other attributes of each face, such as gender, hair color and age[1]. Regarding licensing, FDRI is available under an Apache2 open source license[2]. Finally, although FDRI is herein presented within the context of Autopsy, FDRI can also be run as an independent application.

Currently, *i*) deep CNNs, *ii*) the advent of GPU and *iii*) large datasets of labeled images for training are cited as determinant factors to boost image-based detection and recognition algorithms [28]. In the past, training classical CNNs with multiple layers often failed,

[1] https://azure.microsoft.com/en-us/services/cognitive-services/face/

[2] Anonymous (for double-blind review) versions of FDRI can be found at https://bit.ly/2Oba2va



(a) Microsoft Face API (b) Dlib/FDRI

Figure 1: Face detection by Microsoft Face API (left) and dlib/FDRI (right)

since beyond a certain threshold of stacked layers, the training error would rapidly increase, an issue called the *degradation problem*. This limitation was overcome by resorting to *residual networks*, nicknamed ResNet. Residual networks depart from plain CNNs by focusing on optimizing the layers to fit residual mappings instead of the desired underlying mapping [8]. This approach was pioneered by Hu et. al. and first applied for image recognition in their now seminal work [8], where a 152-layer residual net was successfully used on recognition tasks on ImageNet. A follow up of the pioneer work reached a 200-layer for ImageNet and a 1001-layer ResNet on the CIFAR-10 dataset [9]. All of this has significantly boosted precision in detection and recognition algorithms. As we shall see later on, FDRI also relies on a ResNet-based deep CNN to detect and recognize faces.

2.2 Face Recognition

Sepas-Moghaddam et al. [22] provides a thorough multi-level taxonomy of face recognition, which encompasses *i*) Face structure; *ii*) Feature support; *iii*) Feature extraction support and *iv*) Feature extraction sub-approach. Zeinstra et al. survey forensic facial recognition, pointing out that some scientific literature criticize the precision of facial recognition performed by humans, stressing that it lacks a true scientific approach [29]. Interestingly, Russell et al. [21] coined the term *super-recognizer* to describe persons with above average ability to recognize faces, a skill that is being used by some police forces [17]. Zeinstra et al. also highlight that facial recognition based on deep learning algorithms relies on abstract features that are difficult to understand [29]. Jain et al. review use cases and applicability of automated face matching and retrieval in forensics applications [11]. The main use case focused by Jain et al. is the attempt to identify an individual seen in a frame from a surveillance camera, by looking up for matching faces on available data sources, for example, police mug shots or driving license databases. They argue that due to the poor quality of images sourced from CCTV, social medias and alike, the facial recognition process can only be semi-automated. Additionally, they identify the following four key factors that can derail automated face-recognition accuracy: pose, illumination, expression and aging variation. This way, a prudent approach for face recognition is to have an human in the loop to validate the results. Our use case is more constrained: an expert working on a forensic case, where the analysis of digital still photos is performed either to detect photos where a given individual is present, or at least to get the set of photos where faces were detected, so that only this set has to be manually analyzed by the forensic expert. Videos such as CCTV streams are out of scope of this work.

Mashhadani et al. [16] present the Multimedia-Forensic Analysis Tool (M-FAT). The planned tool aims to combine several image analysis techniques to improve the outcome of automated image processing for digital forensics. Some of the techniques to be combined by M-FAT are automated annotation of images, exclusion of images based on metadata, and the fusion of several cloud-based services to improve object and facial detection. The team of M-FAT study multi-algorithmic fusion approach for face recognition [1]. They experimentally study the results of three commercial systems – Neurotechnology, Microsoft Recognition Service and Amazon Rekognition – with a dataset of real world photos. Their experiments show that merging results from the three systems improves precision, from 67.23% yielded by the best individual system (Microsoft) to 71.60% for the combined systems. These results also show how challenging is facial recognition in real world conditions, since the precision reached by the fused system is barely above 71%. As pointed out by the authors, the usage of cloud-based services raises serious privacy issues. Moreover, cloud-services have costs, which might not be precisely known beforehand. Anda et al. [2] also assess several algorithms and online services for age estimation through face analysis in digital photos. The authors report that for age estimation through face analysis on digital photos, Microsoft's Azure service is the best overall, but all algorithms and services overestimate age for young children and underestimate for aged seniors. Note that age estimation is particularly relevant to detect child abuse material (CAM), an area that is out-of-scope for this work. FDRI relies solely on algorithms that are run locally, requiring for proper execution performance an NVidia's GPU. The rationale behind this approach is to avoid the usage of cloud-based services due to privacy and cost reasons. Indeed, cloud-based service requires uploading the photos, which might violate privacy laws in certain countries. Moreover, cloud services can incur significant costs, in both bandwidth and services.

Face recognition has been adopted in some mainstream devices to grant access to the authorized person. Examples include Apple's FaceID available in iPhone X smartphones and Microsoft's Windows Hello [27]. Both systems rely on special hardware. For instance, the iPhone X uses two frontal cameras plus an infrared camera and a 30 000 dot projector when authenticating with FaceID. Face recognition is also being used for security and commercial purposes, for example, to detect persons of interests at events such as music concerts [19], although when dealing with large crowds or poor lighting, the accuracy is severely diminished[18]. The use of face recognition for authentication in FaceID and other similar technologies has contributed to the debate on the danger of face recognition regarding privacy [15].

3 AUTOPSY SOFTWARE

Autopsy[3] is an open source software for digital forensics available as a desktop GUI application for the main desktop operating systems, i.e., Windows, macOS and Linux. It allows, under a functional graphical user interface, the processing of digital forensic images. For this purpose, it harbors a wide collection of tools for digital

[3] https://www.sleuthkit.org/autopsy/

forensics. For instance, for file recovery and carving, Autopsy interacts with the open source PhotoRec software, while RegRipper, a set of PERL scripts, is used for analyzing Windows registry. Interaction with file systems and files is performed through the dedicated Sleuth Kit framework, while text indexation and search rely on Apache's SOLR. Data and metadata of files are extracted through Tika. Autopsy operates on data sources, with each data sources holding data to be examined. As the name implies, a data source holds data to be examined,with Autopsy supporting several data source formats, such as disk images (e.g, raw, E01, etc.), virtual machines, local disks or a logical set of files.

3.1 Modules in Autopsy

Functionalities can be added to Autopsy through modules. Modules can either be coded in Java or in Jython, a Python version for the JAVA virtual machine. Any module needs to be explicitly ran by the forensic expert. Autopsy has two main categories of modules: *i*) *Ingest* and *ii*) *Report*. Ingest modules are expected to be run when a data source is added (e.g., the E01 forensic image of a hard disk), although they can also be ran at any time. Similarly, while report modules should be run when the data have been analyzed and tagged by the forensic expert, they can be run whenever the forensic expert feels the need to. Ingest modules are further divided in two: *i*) *file ingest* and *ii*) *data source ingest*. The main difference between the two is that a file ingest module has its code triggered for each file existing in a data source, while a data source ingest module receives a reference to a whole data source, leaving the responsibility for the module to find the files that require analysis. Note that any module, independently of its type, can call external applications for specific tasks. As we shall see later on, the FDRI module follows this approach, calling a specially developed external console application – `FDRI.exe` – to perform face detection and recognition tasks.

The installation of a Jython module in Autopsy is straightforward: one just has to create a subdirectory in the module directory of Autopsy and copy into this newly created directory the files that comprise the module, namely the file with the jython code and any auxiliary file. This easiness of installation is another advantage of Jython-based modules.

4 THE FDRI MODULE

The FDRI module is implemented as a data source ingest module. FDRI is comprised of two main components: the Jython code – FDRI.py – and the external executable – FDRI.exe. Within Autopsy, the FDRI module works as follows. First, the forensic examiner activates the module through the *Run Ingest Modules* menu entry in Autopsy, adjusting the execution parameters for the desired execution configuration. For instance, it is possible to select the file formats to analyze, opting either for PNG, JPEG or both.

4.1 Module Execution Workflow

The module starts by copying the files of the selected type(s) of image format(s) to a working directory within the case hierarchy. This step is skipped if the working directory already has the files to analyze from a previous execution. When the copy of the files is terminated, the Jython code writes the JSON file `params.json`. This file holds the execution parameters that convey the options selected by the user. Next, `FDRI.exe` is launched by the Jython code. After having terminated the setup, the Jython code launches `FDRI.exe`. This command line application parses the JSON file and processes the images for the wanted operations: face detection and, if the appropriate configuration is given, face recognition. `FDRI.exe` produces several outputs: *i*) a directory with copies of the images where at least one face was detected, with the faces highlighted by red square boxes; ii) a text file, named `FDRI_faces.txt` that holds, for each image where at least one face was detected, the name of the file, as well as, the XY coordinates of each detected face; *iii*) a XML file with the results of the executions, namely, the image files and coordinates of each detected faces. The XML file follows the Digital Forensic XML (DFXML) format [6]. This allows other forensic tools that also supports DFXML to interpret the results. In fact, `FDRI.exe` can be run as a stand-alone command line console application. Under this mode, execution parameters are passed through a `params.json`-like file, while results delivered through the above indicated files, namely the DFXML one. The current version of `FDRI.exe` is a Windows console application and thus can only be run on Windows operating systems. It has been tested under Windows 10.

The external `FDRI.exe` application performs the bulk of the analysis, that is, face detection and, if activated, face recognition. For face detection, `FDRI.exe` iterates over all the selected image files, applying for each file, a deep CNN-based algorithm described in the next section. When the face detection stage is over, FDRI.exe performs face recognition, but only if requested to do so by the forensic examiner. Activation of the face recognition stage is straightforward: the examiner points out a directory that contains one or more photo(s) of the wanted face(s). This can be done through the GUI configuration interface, or, if `FDRI.exe` is being run independently, through an option specified in the JSON-based parameter file. Either way, `FDRI.exe` uses all image files that exists at the defined directory. We call these images of the wanted individual(s) *positive photos*. If no directory is defined or no image file exists in the directory, then the face recognition stage is simply not ran.

When `FDRI.exe` terminates, control is taken back by the Jython code. The code parses the text result file and register the files where faces were detected. These flagged image files are registered as artifacts in Autopsy. More precisely, these artifacts are grouped into a set called *Images with faces*, which can be browsed by the forensic examiner through Autopsy's GUI. By double clicking on a photo with detected faces, a copy with bounding boxes pointing out the detected faces is shown. Likewise, the files where recognized faces were detected by FDRI are also inserted into Autopsy's blackboard under a group named *WantedFaces*.

4.2 Integration with deep CNN software

FDRI relies on deep convolutional neural networks – DCNN – for detecting and recognizing faces. As stated earlier, it is the external application `FDRI.exe` that first performs face detection and then face recognition. Both features are implemented through the machine learning toolkit dlib [13]. Dlib is a C++ library that comprehends a wide range of software to deal with, among other tasks, numerical, machine learning and image processing algorithms. The

Purpose	File
detection	mmod_human_face_detector.dat
face alignment	shape_predictor_5_face_landmarks.dat
recognition	dlib_face_recognition_resenet_model_v1.dat

Table 1: Data files holding Dlib pretrained models

code of dlib is available under an open source permissive Boost license and includes a wide plethora of examples and documentation. Dlib is developed by Davis King who releases frequent updates. Although other solutions for face detection/recognition exist, dlib was selected due to its ease of programming, its transparent support for NVidia's CUDA, and its high precision in detecting and recognizing faces.

Although `FDRI.exe` chiefly uses the face detection and recognition capabilities of dlib, it also uses some image processing routines from dlib. The images processing routines serve to identify corrupted files, draw bounding boxes on copies of images to highlight detected faces, and to resize photos. The size of an image needs to be within the $[1200 \times 1200, 2500 \times 2500]$ interval. Thus, images smaller or larger than the *normal* interval are downsized or upscaled, respectively. The maximum size 2500×2500 was selected to accommodate any Nvidia's GPU that has at least $2GiB$ of graphical memory.

4.2.1 Face Detection with Dlib. Face detection in dlib is based on a modified Microsoft's ResNet-34 [8]. Instead of the 34 layers of ResNet-34, dlib's CNN has 29 convolutional layers. Another change is the usage of half the filters per layer than the ones used in the original ResNet-34. As reported by Davis King [12], the adapted ResNet model made available by dlib was trained by the author with a dataset of about three million photos with faces, representing 7485 different individuals. The training dataset comprises photos from other datasets and other ones harvested from the Internet.

At the code level, the data of the pretrained models available with dlib are contained in the files listed in Table 1. These files are shipped with FDRI and need to be accessible from the digital forensic machine as they are loaded by `FDRI.exe`. Note that the pretrained models are public domain.

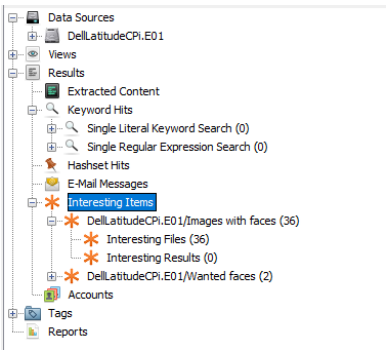


Figure 2: Results of FDRI within Autopsy

Dlib adapted ResNet-29 and the available models have two main limitations: *i*) it cannot detect faces whose size is less than 80×80 pixels; *ii*) the needed processing power increases with the image size. Since dlib has supports for CUDA's cuDNN, execution times can be significantly reduced through the use of a CUDA-supporting GPU. However, the use of a GPU has its own limitations, imposed by the amount of memory of the GPU, which limits the maximum image size that can be processed. For instance, for a GTX 1050 Ti with 2 GiB of GDDR5 memory, the maximum image size that can be processed is 2500×2500. Therefore, images above the memory-imposed threshold are downsized, being successively halved until their dimensions are below the threshold. As we shall see with the experimental results, the downsizing can provoke face detection misses. For example, a face with a square zone of 150×150 pixels in the original photo, can be reduced to half this size – 75×75 pixels – in the downsized version, and thus become too small for the face detector. Images whose dimensions are below the threshold 1200×1200 are successively doubled until the minimum threshold is reached. The upscale operation is done to improve the probability of detecting faces. Note that FDRI allows for different minimum and maximum dimension values, with maximum size being ultimately limited by the amount of graphical memory of the GPU.

4.2.2 Face Recognition with dlib. For face recognition, FDRI follows the approach exemplified by Davis King in his blog [12]. For each detected face, first FDRI uses dlib to extract features through the pretrained models. The next step involves the estimation of the face pose and the needed adjustments for face matching. For this purpose, dlib's default face pose estimator is used to determine five facial pivotal points and then aligned with the face under analysis. The 5-point model is represented by the file `shape_predictor_5` (Table 1). After these steps, a 128-dimension vector is filled with the features extracted from the face, effectively representing the face in a 128-dimension space. Therefore, at the end of this stage, each detected face will be represented by a 128-dimension vector. A 128-dimension vector is also computed for every detected face in the positive photo(s) provided by the forensic examiner and that portrays the individual(s) whose face(s) is sought.

Next, the algorithm enters its matching stage. Specifically, the Euclidean distance is computed between every pair formed by a detected face and a wanted face (*positive face*), one pair at the time. This means computing the Euclidean distance over the two 128-dimension vectors that represent each face of the pair. Pair of faces that are close, that is, which have a small Euclidean distance between them, have a high probability of representing the same face. On the contrary, unrelated or poorly related pair of faces will be separated by a relatively large Euclidean distance. Under FDRI, for the purpose of face recognition, two representations of a face are considered as being the same if the Euclidean distance between them is less than 0.6. This threshold is actually defined by the pretrained model made available by dlib. Changing the threshold would require retraining the face recognition model of dlib [12], a computationally expensive and data intensive operation. Finally, FDRI.exe produces the output files and then terminates. As the final step, the Jython module parses the result files, displaying files with detected faces and files with recognized faces into the corresponding Autopsy interface. As shown later on the computational performance analysis, the face recognition stage is much faster than the face detection stage.

5 EXPERIMENTAL ASSESSMENT

In this section, we assess FDRI and its ability to detect and recognize faces. For this purpose, we analyze the precision and recall for face detection and face recognition with three datasets. One dataset is the well know *Labeled Faces in the Wild* (LFW) [10]. The second one is a digital forensic copy of the hard disk of a daily used personal laptop. We name this dataset *LAP-forensic*. Finally, the third dataset comprises 56 real world unrestrained photos, all with faces. Moreover, 33 of these photos contain the face of a given individual, with some photos also having other faces. We name this third dataset *Unrestrained Faces 56* (henceforth UF56) and use it to assess FDRI ability for face recognition.

We also assess the computational performance of FDRI, using execution times as metric. For this purpose, we use two versions of FDRI: one which solely requires CPU (FDRI-CPU) and the regular version of FDRI that requires an NVidia's GPU. As described next, we use a server and three different NVidia GPUs to measure execution times of FDRI.

5.1 Setup

5.1.1 Hardware. A server normally used for performing digital forensic analysis was used to evaluate FDRI. The main characteristics of the server are: *i*) two-CPU Intel Xeon E5-2620@2.10 GHz, six physical cores per CPU, totaling 12 physical cores; *ii*) 32 GiB of RAM; *iii*) a 4 TB 7200 RPM HDD for storage. To assess the performance impact of GPU, the following NVidia GPUs were separately used: *i*) GTX 750 Ti; *ii*) GTX 1050 TI and *iii*) GTX1070. The main characteristics of the GPUs are given in Table 2.

The OS is Windows 10, version 1803, while Autopsy is version 4.9.1. `FDRI.exe` was linked with dlib's version from April 2018. Regarding NVidia's stack, the driver is version 398.11 with CUDA 9.1.

5.2 Labeled Faces in the Wild

To assess the precision of the detection/recognition models for the LFW dataset, we followed the methodology used in [12]. This methodology defines the following steps: *i*) compares 300 positive photos, that is, photos where the individual is in the photos; then *ii*) compares 300 negative photos, i.e., photos were the individual is not present. The method uses the cross-validation with 10 folds, equivalent to 10 subsets of 600 images each, corresponding to a total of 6000 comparisons.

5.2.1 Precision and Recall. The confusion matrix for face recognition by FDRI over the LFW dataset is given in Table 3. As expected, our results are similar, albeit slightly lower, to the ones announced by Davis King [12]. Specifically, precision was 98.1%, while recall was 99.5%.

5.2.2 Execution times. Execution times obtained with the three GPUs are shown in Table 4 for both the detection and the face recognition stages. Due to its poor performance and the length of the LFW test, the CPU-only version of FDRI (FDRI-CPU) was not run with LFW. As it can clearly be seen in the results, the detection operation is much more computationally demanding than recognition: while detection requires a minimum of around 4560 seconds for the fastest GPU, recognition took less than 12 seconds. Indeed, the face detection stage has to process every image to look for faces, while face recognition only needs to focus on the already detected faces. Another time consuming operation is the image resizing that takes place so that the face detector can receive the image to process within the ideal size range. This operation is done resorting to dlib functions and took a cumulative time of roughly 824 seconds for the 13233 images of LFW. Currently the resizing – upscaling in the case of LFW, since all original images are 250×250 – is performed by the CPU, hence the time spent on resizing does not depend on the underlying GPU. As future work, we aim to speed up the resizing operations by moving them to the GPU.

The execution times highlight the importance of the GPU. While the slowest GTX 750Ti required an average of 15264.247 seconds to execute the face detection stage, the fastest GTX 1070 was more than three times faster, averaging 4558.013 seconds. The middle range GTX 1050 Ti required an average of 8760.820 seconds to complete the face detection stage. Similar performance gaps are observed with the face recognition stage. Clearly, GPUs have a crucial role on the performance of FDRI.

5.3 LAP-forensic Image

LAP-forensic image is a forensic image of a 453 GiB hard disk from a laptop running Windows 10. The forensic image size is 77.6 GiB when compacted in Encase's E01 format. First, LAP-forensic was analyzed with Autopsy to process unallocated space to recover deleted files and partial contents. The resulting forensic image contains 238 345 files, of which 6623 are images encoded either in PNG or JPEG format. However, 119 of these files were not recognized as valid PNG/JPEG files, while 122 had no data, that is, they were 0-byte files. The 0-byte files and the invalid PNG/JPEG are due to failed recovery attempts from unallocated space. Additionally, 962 images were deemed too small either because the file was less than 1025 bytes or at least one of its dimension (width, height) was less than the minimal size of 80×80 pixels required for face detection. Therefore, of the 6623 PNG/JPEG files, 5420 were effectively analyzed by FDRI, while the other 1203 were dropped. The main characteristics of the image files of *LAP-forensic* are summarized in Table 5. Regarding the storage size of the 5420 files, most of them are small. In fact, besides the discarded 962 files, 3191 files range between 1 KiB and 10 KiB. These image files come mostly from the caches of browsers, representing buttons and symbols commonly used in web sites. The distribution of LAP-forensic's images per file size is shown in Table 6.

5.3.1 Precision and Recall for Face Detection. The confusion matrix for FDRI's face detection on the *LAP-forensic* dataset is given in Table 7. FDRI found 251 of the 277 images that contained at least one face. However, it wrongly detected nonexistent faces in six images, hence yielding a precision of 97.67% for positive cases. FDRI also correctly classified 5151 images that had no faces, while wrongly classifying 26 images as having no faces, when they had at least one face, thus yielding a recall of 90.61% for positive cases. For negative cases – images without any face –, FDRI achieved 99.88% precision and 99.50% recall. Note that the high precision and recall for negative cases are also a consequence of the imbalance between images with and without faces.

GPU	# CUDA cores	Memory	Bandwidth (GB/s)	Bus width (bits)
GTX 750 Ti	640	1.5 GiB	86.4	128
GTX 1050 Ti	768	4 GiB	112	128
GTX 1070	1920	8 GiB	253.6	256

Table 2: NVidia's GPUs used in the experimental scenarios

		Reality	
		in photo	**not in photo**
Prediction	**in photo**	*TP: 2943*	*FP: 16*
	not in photo	*FN: 57*	*TN: 2984*

Table 3: Confusion matrix for face recognition in LFW

LFW	Detection (secs)	Recognition (secs)
GTX 750Ti	15264.247	24.719
GTX 1050Ti	8760.820	15.619
GTX 1070	4558.013	11.346

Table 4: Execution times for the LFW dataset

Total files	238 345
Total images	6623
Corrupted images	119
0-byte images	122
Images too small	962
Images to analyze	5420

Table 5: Characterization of *LAP-forensic*

Size of file	# of files
>= 5 MiB	2
[2 MiB, 5 MiB[	35
[1 MiB, 2 MiB[	48
[512 KiB, 1 MiB[	39
[100 KiB, 512 KiB[	217
[50 KiB, 100 KiB[	322
[40 KiB, 50 KiB[	152
[30 KiB, 40 KiB[	231
[20 KiB, 30 KiB[	457
[10 KiB, 20 KiB[	722
]1 KiB, 10 KiB[	3191

Table 6: Size of image files of *LAP-forensic*

Of the 26 images mistakenly classified by FDRI as having no faces, 14 had faces whose dimensions in the image were less than 80×80 pixels, that is, too small to be flagged by the detector. The remainder 12 were from photos taken with poor illumination and noisy backgrounds of individuals engaged in sporting activity, and thus not posing for the camera, that is, not looking at the camera. An additional common property of these 12 photos was that their dimensions: they all were 3872×2592 pixels. This resolution is above

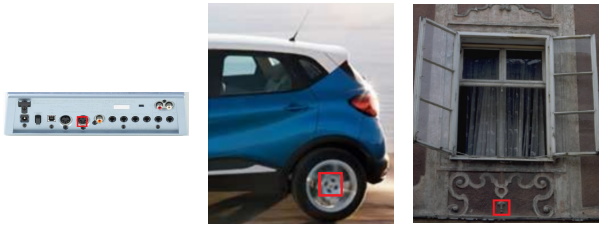


Figure 3: False positive photos reported by FDRI in LAP-forensic

the top 2500×2500 threshold, and thus the photos were downsized prior to face detection. In this particular case, each 3872×2592-sized photo was halved in each dimension to 1936×1296. This has the negative effect of also reducing the size of existing faces, effectively halving them, possibly below the 80×80 threshold. To assert this hypothesis, we reran the face detection algorithm with each of the 12 photos divided into four equally-sized parts. That is, instead of letting the algorithm automatically halve the photos, we provided as input, for each of the 3872×2592-sized photos, the four 1936×1296 images obtained by splitting the image into four equal parts. In three of the 12 photos, this allowed the algorithm to detect faces that were not previously found. We plan to pursue this venue in future work for large resolution images. Overall, and not considering the four-part divisions, the detection algorithm downsized 35 images, upscaled 5294, leaving 91 untouched. On average, the upscaling took a total of 396.265 seconds, while the downsizing operations were performed in 5.054 seconds.

Of the six *false positives* of FDRI, two of them were wrongly flagged as having a face because an Asian alphabet symbol represented inside a circle was erroneously reported as a face. Another wrongly reported *false positive* face was a cross symbol contained within a circle. Note that these three misclassified occurred on synthetic images. The other three wrong predictions occurred with optical photos: one of a sound equipment where a set of buttons was mistaken for a face, another one of a moving car, where the spinning back wheel was incorrectly classified as a face, and finally, one where the ornament part below an ancient windows was mistaken for a face. These false positives are shown in Figure 3, with the falsely predicted face highlighted for better visibility.

5.4 Execution Times

Execution times for face detection in the *LAP-forensic* dataset are shown in Table 8. Clearly, a GPU is indispensable to achieve proper performance. Indeed, the CPU-only (FDRI-CPU) execution took 74 265.356 seconds, that is, more than 20 hours, while the slowest GPU – GTX 750Ti – was 15.5× faster, requiring 2097.9 seconds to execute the face detection stage. Comparatively to the fastest GTX

		Reality	
		in photo	**not in photo**
Prediction	**in photo**	*TP: 251*	*FP: 6*
	not in photo	*FN: 26*	*TN:5151*

Table 7: Precision of face detection on the LAP-forensic dataset

LAP-forensic	**Detection (secs)**	**Recognition (secs)**
CPU	74265.356	N.A.
GTX 750Ti	4740.153	N.A.
GTX 1050Ti	3343.647	N.A.
GTX 1070	2097.9	N.A.

Table 8: Execution times for the *LAP-forensic* dataset

1070, FDRI-CPU was roughly 35× slower. Execution times achieved by the GPUs follow their performance, with the GTX 1070 being approximately 2.25× faster than the GTX 750 TI.

5.5 UF56

The main characteristic of the UF56 dataset is that is solely comprised of unrestrained real world photos. In these real world photos, the facial pose, illumination, accessories – glasses, caps, bike helmet, etc. – of the individuals, as well as the background, are unrestrained, as it happens in regular photos. Additionally, some photos hold multiple faces. In summary, *UF56* is comprised of photos that are found in digital forensic analysis.

The face recognition performance of FDRI on the *UF56* dataset was assessed with five different *positive* photos of an individual. The individual was present in 33 photos of the dataset. The main characteristics of the five photos are summarily given in Table 9, while the photos are shown in Figure 4. For the sake of simplicity, we name them *photo #1* to *photo #5*. FDRI was first run separately for each of the five positive photos, and then, run with all five photos given at once. The precision results are shown in Table 10.

Photo #2, which is from a document ID, yielded the worst precision results. This is surprising since the photo was taken by a document identification service, with specific equipment and following the quite strict protocol for such photos. For instance, the individual needs to be directly facing the camera and to present a neutral pose for the photo (e.g., she/he cannot smile). Additionally, accessories such as glasses, cap, or any other type of hat are not allowed. A possible explanation is that the photo of the document ID, although optimal for recognition under specific conditions – illumination, front pose, etc. – is less appropriate for *real world* photos. Indeed, in *real world* photos, the individual might not be posing for the photo and might be captured by the camera from non-direct angles (e.g., *side photo*). Besides, the individual might be using accessories such as glasses (corrective or sunglasses), wearing a cap or even a bike helmet. Illumination might also be non-ideal. For instance, in the *UF56* dataset, three photos where the wanted individual appeared were taken against sunlight, yielding dark photos. *Photo #2* also triggered five false positives, while the other *positive*

Positive photos	**Size**	**Size of face**	**Description**
#1	1198x898	487x629	Regular
#2	193x254	133x179	Document ID
#3	1920x2560	661x749	With hat and sunglasses
#4	2048x1536	153x87	With sunglasses
#5	1440x1920	453x608	With glasses

Table 9: Size and main characteristics of the *positive photos*

Recognition	**True positives**	**False negatives**	**False positives**
Photo #1	22 (67%)	11	1
Photo #2	19 (58%)	14	5
Photo #3	22 (67%)	11	1
Photo #4	22 (67%)	11	1
Photo #5	21 (64%)	12	1
Overall	27 (82%)	6	6

Table 10: Precision per *positive photo* for FDRI on *UF56*

(a) #1 **(b) #2** **(c) #3** **(d) #4** **(e) #5**

Figure 4: Positive photos of the sought individual in *UF56*

photos triggered one false positive each. Note that contrary to false negatives, false positives can be considered as a minor nuisance, since every result classified as positive always needs to be review by an human expert. Another interesting issue happened with *Photo #3*: FDRI was able to recognize the wanted face in all but one photo where the individual was either wearing a cap or a cyclist helmet, yielding the best performance for this type of photos. This occurred despite the hat of *Photo #3* being substantially different than the caps – they were two different caps – and obviously of the cyclist helmet.

Overall, the best recognition result was achieved with all five *positive photos* given at once, yielding a precision of 82%, while the highest precision achieve with any single photo was 73%. Note that using five *positive photos*, only marginally increases the computational costs and henceforth the execution time of the recognition stage. Therefore, whenever possible, FDRI should be provided with a meaningful set of different photos of the wanted individual(s).

5.6 Execution Times

The execution times for *UF56* are shown in Table 11. Comparatively to the two other datasets, execution for *UF56* are faster simply because of the lower number of images of this dataset.

UF56	Detection (secs)	Recognition (secs)
CPU	1473.237	7.346
GTX 750Ti	405.089	2.185
GTX 1050Ti	288.109	1.421
GTX 1070	180.136	0.993

Table 11: Execution times for *UF56* dataset

Although the trend is less visible than the one observed with the LFW dataset, the GPU still heavily influences the execution time. Indeed, the GTX 1070 is roughly 2.2× faster than the GTX 750 Ti and almost 1.6× faster than the GTX 1050 Ti. As previously observed with LAP-Forensic, FDRI-CPU is no match for any of the GPU, being 8× slower than the fastest GPU. Finally, the face recognition stage was significantly faster than it had been for the LFW dataset, again because of the lower number of faces per photo that exists in *UF56* comparatively to the LFW dataset. The speed difference among the GPU is again reflected in the execution times for face recognition.

5.7 Limitations

A major limitation of Dlib is ethnic bias, with the model yielding poor recognition results with non-Caucasian faces, especially with Asian individuals. This an acknowledged problem of Dlib caused by the limited diversity of faces used to train Dlib's model [5]. To evaluate the performance regarding Asian faces in unrestrained photos, we performed a test with the 50 photos of "AP Photos: Editors' picks for 2018 from Asia"[4]. The face detection test correctly identified all the photos that had at least a visible face, except for photo #35 (the photos are numbered from 1 to 50). In photo #35, the only visible face is in a left profile position, although clearly visible. To evaluate the ability of FDRI to recognize Asian faces, we searched for a face that does not exist in the 50-photo dataset. This way, any match was a false positive. We used as positive photos, two of the pictures of the Wikipedia's page of actor Jackie Chan[5]. The recognition algorithm erroneously reported matches in the following eight photos: #3, #4, #7, #18, #19, #26, #46 and #48. Although the test has a limited range since the dataset has solely 50 photos, it does hint that the used Dlib's model is inappropriate for recognition of Asian faces. We plan to further study this issue in future work.

6 CONCLUSION

We presented the FDRI software that provides facial detection and recognition in photos of digital forensic cases. Specifically, FDRI aims to *i*) detect all faces existing in digital images and *ii*) signal the images where the face of a given sought individual(s) is recognized. Although FDRI is integrated with the digital forensic software Autopsy, it can also be run separately or integrated within another digital forensic software.

Experimental assessments show that FDRI can yield high precision face detection, provided that faces in the photos are larger or equal to 80×80 pixels after the images have been normalized. Normalization of an image is performed by the 1^{st} stage of FDRI and simply consists in downsizing or upscaling the image depending on its original size. Other factors such as illumination of the image and face orientation also contribute to the precision quality of face detection.

Experimental results also show that the precision of face recognition is highly dependent on the variety and quality of the provided *positive photos*. Indeed, when used individually, the highest true positive precision achieved was 67.67%, while the simultaneous usage of the five *positive photos* yielded 81.82%. Additionally, using photos for document ID such as *citizen card ID*, passports and driver licenses might yield low precision results when the dataset to search is comprised of regular ad-hoc photos. Overall, despite all progresses, face detection and face recognition are still far from perfect in unrestrained photos and thus results always need the careful revision of appropriate experts.

The execution time of FDRI is mostly dominated by the face detection stage. For a proper execution speed, FDRI requires A CUDA-aware GPU, with the execution performance being highly dependent on the underlying GPU, namely for face detection. For instance, experimental assessments showed that a GTX 1070 provided for an execution roughly thrice faster than a GTX 750 Ti. As the GPU performance evolution trend is expected to continue in the years ahead, we expect that FDRI will benefit, achieving a lower execution time per processed image.

As future work, we plan to optimize FDRI to reduce its execution times, namely by moving the image resize operation to the GPU. We also aim to study strategies that can increase the precision detection when large images are processed. Right now, due to downscaling of these images performed by the normalization stage, some faces might be reduced to dimension less than the 80×80 pixels thresholds, and thus be missed by FDRI. Finally, we also plan to test alternative face detection and recognition algorithms, assessing their accuracy, especially when dealing with non Caucasian faces.

ACKNOWLEDGMENTS

This work was partially supported by FCT and Instituto de Telecomunicações under project UID-EEA-50008-2013 and by DEI-ESTG-Polytechnic Institute of Leiria.

REFERENCES

[1] Hiba Al-Kawaz, N Clark, Steven M Furnell, Fudong Li, and A Alburan. 2018. Advanced facial recognition for digital forensics. In *17th European Conference on Cyber Warfare and Security*. Academic Conferences and Publishing International Limited, 11–19.

[2] Felix Anda, David Lillis, Nhien-An Le-Khac, and Mark Scanlon. 2018. Evaluating Automated Facial Age Estimation Techniques for Digital Forensics. In *12th International Workshop on Systematic Approaches to Digital Forensics Engineering (SADFE), IEEE Security & Privacy Workshops*. IEEE.

[3] Doug Beaver, Sanjeev Kumar, Harry C Li, Jason Sobel, Peter Vajgel, et al. 2010. Finding a Needle in Haystack: Facebook's Photo Storage.. In *OSDI*, Vol. 10. 1–8.

[4] Doug Beaver, Sanjeev Kumar, Harry C Li, Jason Sobel, Peter Vajgel, et al. 2016. FaceRadar: Extending Open Source Software to Accelerate Image Processing in Digital Forensic Investigations through Face Detection. In *The Symposium on Computing at Minority Institutions*.

[5] dlib. 2018. Improve dlib (dlib_face_recognition_resnet_model_v1) with Asian faces. Website (access on 2019-03-20). https://github.com/davisking/dlib/issues/1407.

[6] Simson Garfinkel. 2012. Digital forensics XML and the DFXML toolset. *Digital Investigation* 8, 3-4 (2012), 161–174. https://doi.org/10.1016/j.diin.2011.11.002

[4] https://www.apnews.com/3f9f68299d4e42fa86027e45ee8bdff1

[5] https://en.wikipedia.org/wiki/Jackie_Chan

[7] Simson L Garfinkel. 2010. Digital forensics research: The next 10 years. *digital investigation* 7 (2010), S64–S73. https://doi.org/10.1016/j.diin.2010.05.009

[8] Kaiming He, Xiangyu Zhang, Shaoqing Ren, and Jian Sun. 2016. Deep residual learning for image recognition. In *Proceedings of the IEEE conference on computer vision and pattern recognition*. 770–778. https://doi.org/10.1023/B:VISI.0000013087.49260.fb

[9] Kaiming He, Xiangyu Zhang, Shaoqing Ren, and Jian Sun. 2016. Identity Mappings in Deep Residual Networks. In *Computer Vision – ECCV 2016*, Bastian Leibe, Jiri Matas, Nicu Sebe, and Max Welling (Eds.). Springer International Publishing, Cham, 630–645. https://doi.org/10.1007/978-3-319-46493-0_38

[10] Gary B Huang, Marwan Mattar, Tamara Berg, and Eric Learned-Miller. 2008. Labeled faces in the wild: A database for studying face recognition in unconstrained environments. In *Workshop on faces in 'Real-Life' Images: detection, alignment, and recognition*.

[11] Anil K Jain, Brendan Klare, and Unsang Park. 2012. Face matching and retrieval in forensics applications. *IEEE multimedia* 19, 1 (2012), 20. https://doi.org/10.1109/MMUL.2012.4

[12] Davis King. 2017. High Quality Face Recognition with Deep Metric Learning. Website (accessed on 2019-03-10). http://blog.dlib.net/2017/02/high-quality-face-recognition-withdeep.html.

[13] Davis E King. 2009. Dlib-ml: A machine learning toolkit. *Journal of Machine Learning Research* 10, Jul (2009), 1755–1758.

[14] Haoxiang Li, Zhe Lin, Xiaohui Shen, Jonathan Brandt, and Gang Hua. 2015. A Convolutional Neural Network Cascade for Face Detection. In *The IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*.

[15] Jennifer Lynch. 2018. Face Off:Law Enforcement Use of Face Recognition Technology. Website (accessed on 2019-03-11). https://www.eff.org/wp/face-off.

[16] S. Mashhadani, H. Al-kawaz, N. Clarke, S. Furnell, and F. Li. 2017. A novel multimedia-forensic analysis tool (M-FAT). In *2017 12th International Conference for Internet Technology and Secured Transactions (ICITST)*. 388–395. https://doi.org/10.23919/ICITST.2017.8356429

[17] Alex Moshakis. 2018. Super recognisers: the people who never forget a face. Website (access on 2019-03-10). https://www.theguardian.com/uk-news/2018/nov/11/super-recognisers-police-the-people-who-never-forget-a-face.

[18] BBC News. 2018. Police facial recognition 'needs considerable investment'. Website (access on 2019-03-12). https://www.bbc.com/news/uk-wales-46359789.

[19] BBC News. 2018. Were Taylor Swift fans tracked at her gig? Website (access on 2019-03-12). https://www.bbc.com/news/technology-46567125.

[20] Darren Quick and Kim-Kwang Raymond Choo. 2014. Impacts of increasing volume of digital forensic data: A survey and future research challenges. *Digital Investigation* 11, 4 (2014), 273–294. https://doi.org/10.1016/j.diin.2014.09.002

[21] Richard Russell, Brad Duchaine, and Ken Nakayama. 2009. Super-recognizers: People with extraordinary face recognition ability. *Psychonomic bulletin & review* 16, 2 (2009), 252–257.

[22] Alireza Sepas-Moghaddam, Fernando Pereira, and Paulo Lobato Correia. 2019. Face Recognition: A Novel Multi-Level Taxonomy based Survey. *arXiv preprint arXiv:1901.00713* (2019).

[23] C. Shu, X. Ding, and C. Fang. 2011. Histogram of the oriented gradient for face recognition. *Tsinghua Science and Technology* 16, 2 (April 2011), 216–224. https://doi.org/10.1016/S1007-0214(11)70032-3

[24] Paul Viola and Michael J Jones. 2004. Robust real-time face detection. *International journal of computer vision* 57, 2 (2004), 137–154. https://doi.org/10.1023/B:VISI.0000013087.49260.fb

[25] Daisuke Wakabayashi. 2013. The point-and-shoot camera faces its existential moment. *Technology* 10 (2013), 59.

[26] Michael J Wilber, Vitaly Shmatikov, and Serge Belongie. 2016. Can we still avoid automatic face detection?. In *Applications of Computer Vision (WACV), 2016 IEEE Winter Conference on*. IEEE, 1–9. https://doi.org/10.1109/WACV.2016.7477452

[27] John Wojewidka. 2017. Why the mobile biometrics surge demands true liveness. *Biometric Technology Today* 2017, 10 (2017), 8–11.

[28] Stefanos Zafeiriou, Cha Zhang, and Zhengyou Zhang. 2015. A survey on face detection in the wild: Past, present and future. *Computer Vision and Image Understanding* 138 (2015), 1–24.

[29] CG Zeinstra, D Meuwly, A Cc Ruifrok, R Nj Veldhuis, and LJ Spreeuwers. 2018. Forensic face recognition as a means to determine strength of evidence: a survey. *Forensic Sci. Rev* 30, 1 (2018), 21–32.