# FaceRadar: Extending Open Source Software to Accelerate Image Processing in Digital Forensic Investigations through Face Detection

Blaize K. Strothers
Faculty Advisor: Dr. Chutima Boonthum-Denecke
Department of Computer Science
Hampton University
Hampton, VA 23668
blaize.strothers@gmail.com

## ABSTRACT
In digital forensics, image analysis represents one of the most labor-intensive tasks during an investigation. However, there is an absence of publicly available, low-cost solutions for streamlining these investigations through image processing automation. Several methods of automated forensic image processing use computer vision to carry out face detection and recognition, but few make use of freely available and extensible open source software. This paper proposes the modification of open source forensic software to add functionality that optimizes forensic image processing through face detection and face recognition add-ons. An original extension module for the Autopsy open source forensic software was developed using OpenCV open source computer vision libraries to accomplish face detection. FaceRadar was created using original design, implementation, and testing efforts. This open source approach to optimizing image processing in digital forensic investigations creates opportunities for image processing at low time, effort, and financial cost, making efficient image analysis in computer forensics investigations possible in a variety of limited-resource settings.

## CCS Concepts
• **Applied computing~Evidence collection, storage and analysis**
• *Computing methodologies~Computer vision problems*.

## Keywords
Digital forensics; open source; computer vision; face detection; FaceRadar.

## 1. INTRODUCTION
In an age in which technology such as computers, cameras and mobile devices are both ubiquitous and constantly evolving, digital forensics plays a crucial role in procuring evidence for criminal investigations. The same devices used for innocuous legitimate purposes are misused to surreptitiously carry out illegal acts, maintain crime networks, and participate in illicit behaviors.

Digital forensic analysis software allows investigators to piece together critical evidence in criminal cases involving the unique challenges posed by modern technology and its inherently complex data lifecycles. However, digital forensics has long relied almost exclusively on closed-source software tools [1]. This means that forensic examinations that depend on data retrieved by closed source tools are based on faith in the tools' accuracy and effectiveness, rather than trust earned by comprehension of the tools' methods, which can only be derived from a transparent relationship with the source code and its inner workings [1].

Open source digital forensics software helps to shape future computer forensic investigations by providing an avenue for the creation of meaningful extension modules that meet the needs of digital forensic examiners, while endeavoring to maintain the quest for truth that drives digital forensics investigations. As picture analysis constitutes a tedious process usually demanding intensive manual forensic processing [2], one of the most critical needs within digital forensic investigation is a reduction in the time and effort required for law enforcement to analyze forensic images.

In creating FaceRadar, an extension module for Autopsy forensics software, open source software is used to strike a balance between the pressing needs for both transparency and efficiency in digital forensic investigations.

## 2. BACKGROUND
### 2.1 Digital Forensic Investigation
In a digital forensic investigation, evidence is derived from digital sources for verification, examination, record keeping, and dissemination that makes easier the reconstruction of criminal events or facilitates the anticipation of planned criminal operations [1]. In this way, digital forensics procedures are applied to computer-generated data sources to produce evidence for a wide range of investigation types [1]. The recovery of system artifacts created due to user actions, such as logs, time stamps and file deletion records, and system actions, such as changes in memory blocks and other automatic processes, contribute to the main goal of a digital forensic investigation, which is to locate facts that prove true a series of events by finding traces of those events left behind on a system [1]. While the main goal of the digital forensic investigator is to uncover the truth within a supposed series of events to prove or disprove the hypothesis of the case, oftentimes the role of the investigator requires an acceptance or rejection of digital evidence in light of the methods and systems that allegedly produced it, with the establishment of the legitimacy of the evidence sometimes being the only reason for an investigation [1].



*ADMI 2016*, March 31-April 3, 2016, Winston-Salem, NC, USA.

## 2.2 Manual Image Processing

Within digital forensic investigations, image processing represents a significant portion of the time required to establish evidence. Computer forensic investigators often manually examine large numbers of images during the search for potential evidence, which requires a formidable amount of scanning and analysis in far-reaching investigations [3]. Chen et al. further enumerated the difficulties of this task when quantifying the challenges involved in a typical image processing segment of a digital investigation – approximately 34,000 browser cache image files on a lab computer with a 6GB hard drive were calculated to require multiple hours of monotonous analysis if the examiner uses only milliseconds to assess each image [3].

The rapidly decreasing cost of data storage and increasing availability in storage device size – electronic searches revealed that a 1TB external hard drive currently costs less than $48 – only exacerbates the laboriousness of the work by further eliminating any reason for users to erase any of their stored images [3]. Chen et al. ascertained that an average computer user could be expected to store hundreds of thousands of images, and in the same vein, that a commercial target such as a web hosting service might contain tens of millions of images, creating an exceedingly difficult task for investigators and requiring that the search be narrowed [3].

## 2.3 Automated Image Processing

The automation of image processing and analysis within computer forensic investigations alleviates digital forensic examiners of the excessive time required to manually sort through and analyze images to determine their relevance and possible value to a case. However, not every form of automated image processing guarantees a level of analysis comparable to that achieved through manual processing.

James and Gladyhsev demonstrated that a comparison of automated forensic analysis with manual analysis exhibited that automated forensic triage produced results comparable to manual triage in only a few cases that did not require specialized knowledge, such as detection of images of exploited children, and automated analysis was even less reliable in more involved cases requiring in-depth knowledge of the system [4]. While James and Gladyhsev assert that digital forensic examiner preparation level, the possibility of overlooked evidence, and issues of evidence verification pose specific challenges to automated forensic examination [4], in many cases automated image processing still lessens the workload on examiners while producing at least marginally viable results that can be refined in a much smaller amount of time.

## 2.4 Open Source Software in Digital Forensics

Open source tools aid digital forensic investigations namely through their source code provision – access to source code allows examiners to review and change it to suit their needs, as well as verify their evidence by providing as proof the source code from which the data was produced in the case of false negatives or other erroneous results [1]. Open source forensic tools aid digital forensic examiners in their quest to uncover the truth by providing a level of transparency closed-source tools rarely, if ever furnish – the open source forensic toolkit and Autopsy predecessor the Sleuth Kit provides at least three separate methods of bug tracking: change logs provided with each release, public bug trackers on its project site, and actual source code build comparisons, allowing examiners to validate any code changes themselves instead of relying on and trusting the software distributor [1].

Open source tools are also generally free of cost, and while the financial benefits support students and burgeoning digital forensic examiners, they also greatly aid digital forensics departments with restricted budgets, as well as those with a full set of commercial, closed-source tools looking to expand their forensic toolkits at low to no cost. The low cost benefits countries with limited budgets, which can use open source software to access modern evidentiary technology without needing to spend time or financial resources not possessed. The portability and flexibility of open source forensics tools means that they can be easily moved among systems without expensive proprietary licenses and can also be used as needed, with a wide range of available installation options, none of which require purchase orders, copy protection, or software provider permissions or licenses [1]. This means that computer forensic investigations can take place on multiple computer systems or networks without the hassle of or consideration for licensing or install stipulations.

In addition, open source tools allow would-be examiners to learn from forensic analysis tools by inspecting the options with which the tool was executed, the tool output and the code that produced it, allowing them to better understand how the tool functions [1]. Open source programs are also often accompanied by committed communities of users and developers willing to assist with problems or inquiries [1].

# 3. IMPLEMENTATION

## 3.1 Open Source Computer Vision

The Intel Open Source Computer Vision (OpenCV) libraries provide cross-platform programming functions that perform real-time computer vision under the open source BSD license. As seen in Viola and Jones [5], OpenCV face detection using Haar cascades provides accurate detection rates at high speeds based on detection algorithms that iterate over images in multiple stages of Haar-like feature composition using a cascading design that implements “strong classifiers” [6].

## 3.2 Using Haar-like Features

Implementing face detecting through the OpenCV Haar cascade classifier allows for the identification of Haar-like features to find faces using an OpenCV cascade classifier, which are classes that can be trained to recognize a type of image through positive and negative examples, and which are then “cascaded” by being applied successively to an image until the image is accepted or fails to fulfill a classifier stage [7].

Haar-like features are used as input to classifiers and produce detection results through their shapes, placement within the image, and scale [7]. An image detection result is calculated by placing the rectangular features anywhere on the image in question, and finding the difference of the sum of image pixels under the positive (black) and negative (white) areas of the rectangle, thus verifying the existence or absence of image characteristics such as edges, lines, and center-surrounded features as shown in Figure 1 [7].

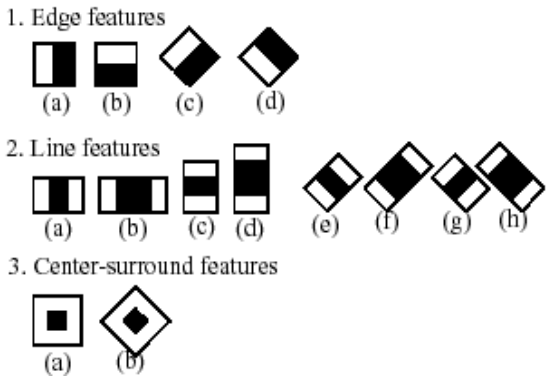


**Figure 1. Haar-like features [7].**

## 3.3 Creating the Autopsy Module, FaceRadar

Autopsy is a digital forensics program and the graphical user interface to the Sleuth Kit command line forensics framework [8]. FaceRadar was created as an Autopsy file-level ingest module that processes images within an Autopsy case using OpenCV Haar cascades and displays detected faces within the program main window. FaceRadar detects image files through an algorithm developed by Rajmund Witt that identifies .jpeg, .bmp, .png, .gif, and .tiff file headers. The module then scans each image for faces using the OpenCV Haar cascade.

The FaceRadar module was implemented using Autopsy ingest module documentation. The module checks the type of each file used as input for the ingest module, checks the user preference for skipping any known files, then checks if the input file is an image file and if so, attempts face detection and creates an array of results to post to the Autopsy Blackboard, which is the left side panel containing the FaceRadar Detected Faces section that can be seen in Figure 4. Figure 3 shows the file processing flow of the FaceRadar ingest module.

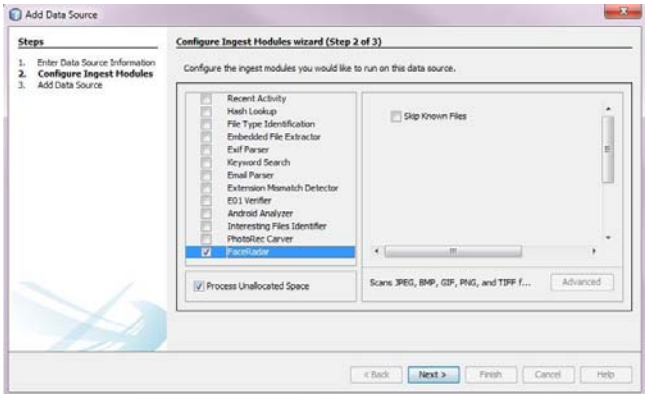


**Figure 2. FaceRadar ingest module during Autopsy case creation.**

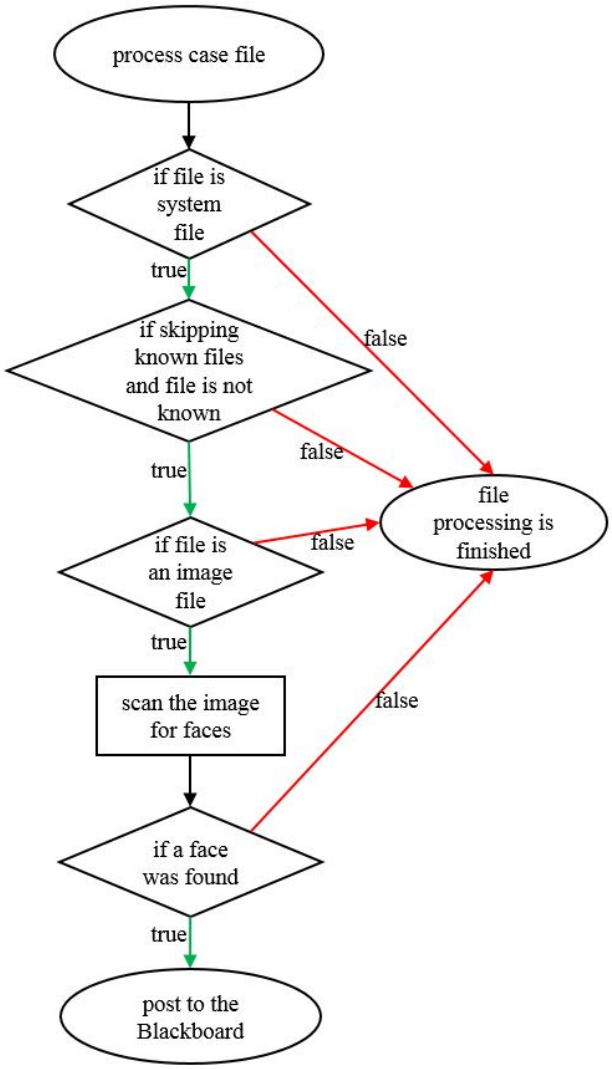


**Figure 3. FaceRadar application flowchart.**

## 4. RESULTS

During the first test, a set of test images was created from a randomly selected 116-image subset of over 2000 images from 101 different subject categories [9]. FaceRadar identified all 17 images of human faces, but also identified faces in two photos focused on other subjects. In a second test using 91 images from the same original set along with 17 text files, an .mp3 song, a .mobi electronic book, and a .tmp temporary file, FaceRadar selected only the images and then detected all 7 faces in the set, as well as a drawing of a face and an erroneous result. Additional tests were then conducted to determine the accuracy of the module.

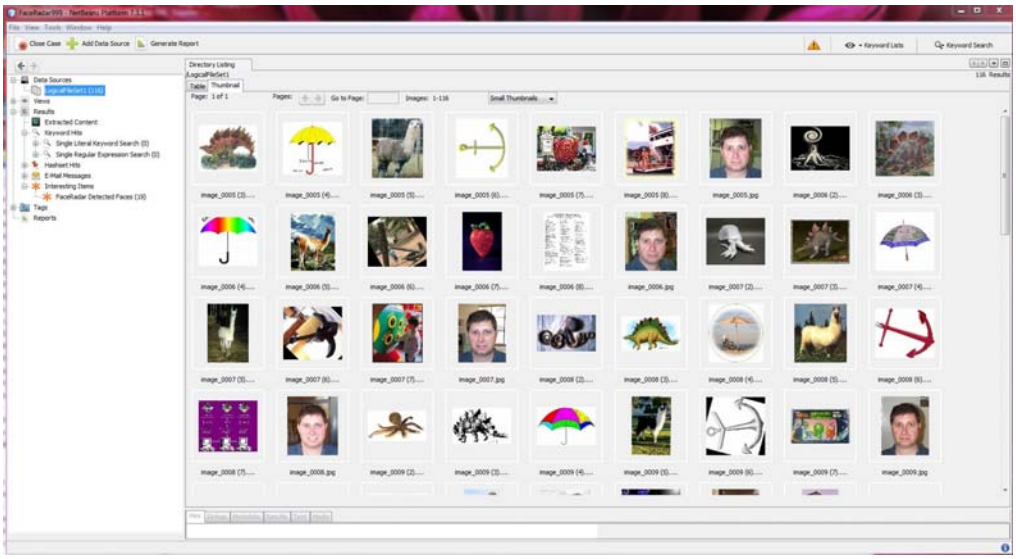

**Figure 4. Set of test images within Autopsy processed by the FaceRadar module during the initial test.**

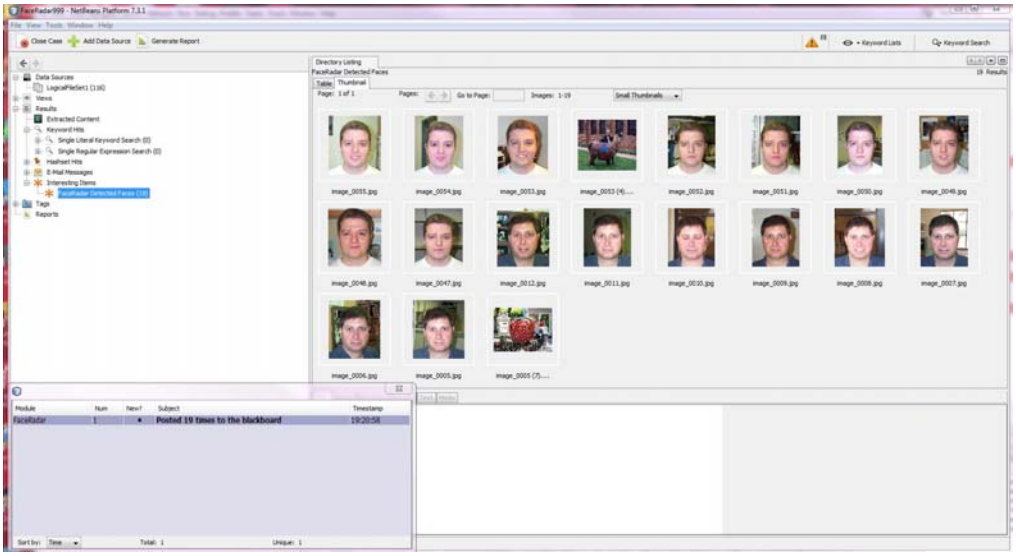

**Figure 5. Faces detected by the FaceRadar module within Autopsy during the initial test.**

## 4.1 FaceRadar Precision and Recall Statistics

The FaceRadar module precision was calculated using the information retrieval definition of precision, in which precision is the fraction of retrieved set elements that are relevant to the search. Precision was documented over a series of 10 test runs with image sets that varied in the number of total images and the proportion of relevant to non-relevant images, as well as the specific images in each set – both relevant and non-relevant images were selected at random. Precision was calculated according to the following equation [10]:

$$\text{precision} = \frac{\text{number of relevant items retrieved}}{\text{total number of items retrieved}}$$

The average precision for FaceRadar was 0.90, meaning that most, but not all of the images retrieved by the module were relevant, or contained faces.

The information retrieval definition of recall, in which recall is the fraction of the relevant search elements that are retrieved, was used to determine the recall of FaceRadar. The same 10 test image sets of varying sizes were used to calculate precision were used to calculate recall and the results were documented in the statistics summary table in Figure 6. Recall was calculated using the following formula [10]:

$$\text{recall} = \frac{\text{number of relevant items retrieved}}{\text{number of relevant items in collection}}$$

The average recall for FaceRadar was 0.99, meaning that almost all relevant images were retrieved in each test case, or that each

designated face image was usually returned as a hit within the Autopsy Interesting Items section under "FaceRadar Detected Faces."

Within the Autopsy FaceRadar Detected Faces hit results, non-relevant images retrieved often included semblances of faces such as the image in the center of U.S. paper currency, high-contrast animals' faces (such as the faces of Dalmatians and pandas), and statues. This contributed to the reduced precision scores, as these images were not designated relevant face-containing images, but with the exception of the animals, did in fact contain human faces. Future tests with different data sets will use better non-relevant images that exclude all forms of faces.

**Table 1. FaceRadar statistics summary table.**

| Images | Relevant Images | Relevant Results | Retrieved Results | Precision | Recall | F-score |
|---|---|---|---|---|---|---|
| 10 | 3 | 3 | 3 | 1.0000 | 1.0000 | 1.0000 |
| 50 | 15 | 15 | 16 | 0.9375 | 1.0000 | 0.9677 |
| 90 | 30 | 29 | 31 | 0.9355 | 0.9667 | 0.9508 |
| 130 | 45 | 44 | 46 | 0.9565 | 0.9778 | 0.9670 |
| 170 | 60 | 59 | 61 | 0.9672 | 0.9833 | 0.9752 |
| 210 | 120 | 119 | 120 | 0.9917 | 0.9917 | 0.9917 |
| 250 | 60 | 60 | 65 | 0.9231 | 1.0000 | 0.9600 |
| 290 | 8 | 8 | 17 | 0.4706 | 1.0000 | 0.6400 |
| 330 | 100 | 98 | 101 | 0.9703 | 0.9800 | 0.9751 |
| 370 | 80 | 79 | 87 | 0.9081 | 0.9875 | 0.9461 |
| | | | **Averages** | **0.9061** | **0.9887** | **0.9375** |

# 5. DISCUSSION

The creation of the FaceRadar Autopsy extension module establishes proof of concept for the use of open source software extensions to optimize both forensic image processing and digital forensic investigations in general. In particular, this approach is significant due to its use of purely open source software and algorithms, with preliminary results supporting the feasibility of using entirely open source software during digital forensic investigations [1].

Extending open source digital forensics software to make computer forensic investigations more efficient not only provides a significant benefit for examiners lacking time resources, but also those lacking the funds for multiple closed-source software products. By extending open source forensic programs, digital forensics examiners can generate the functionality needed to optimize their investigations, replacing the need for time, money, and other resources not previously available.

# 6. FUTURE WORK

Plans to expand the FaceRadar face detection Autopsy module to include face recognition capabilities are underway, with eigenvectors being considered for face recognition implementation by comparing facial features to sets of known individuals. Future work will modify FaceRadar to allow users to locate pictures of a specific person by using face recognition to match an uploaded image to faces detected on a drive image. While face recognition presents unique challenges when faced with factors such as low image quality, subject position, lighting, facial expression and differences in age [11], further research will be conducted into established and new ways of addressing these complexities.

# 7. ACKNOWLEDGMENTS

This work is partly supported by the National Science Foundation CyberCorps: Scholarship for Service program under grant no. DGE-1303409 (PI/co-PI: Dr. Chutima Boonthum-Denecke, Dr. Jean Muhammad).

# 8. REFERENCES

[1] Altheide, C., and Carvey, H. A. 2011. Digital Forensics with Open Source Tools. Burlington, MA: Syngress.

[2] U.S. Department of Homeland Security (DHS). 2015. Cyber Forensics. Retrieved from http://www.dhs.gov/science-and-technology/csd-forensics

[3] Chen, Y., Roussev, V., Richard III, G.G. and Gao, Y. 2005. Content-based image retrieval for digital forensics. *Proceedings of the International Conference on Digital Forensics (IFIP 2005).*

[4] James, J., & Gladyhsev, P. 2013. Challenges with automation in digital forensic investigations.

[5] Viola, P., and Jones, M.J. 2004. Robust Real-Time Face Detection. *International Journal of Computer Vision 57(2),* 137-54.

[6] Jimenez, A. E. et al. 2015. Tag Detection for Preventing Unauthorized Face Image Processing. In Y. Shi, et al., (Eds.), *Digital-Forensics and Watermarking: 13th International Workshop, IWDW 2014, Taipei, Taiwan, October 1-4, 2014: Revised Selected Papers*. Switzerland: Springer, pp. 513-24.

[7] OpenCV dev team. 2015. Cascade Classification. Retrieved from http://docs.opencv.org/2.4/modules/objdetect/doc/cascade_classification.html

[8] Carrier, B. 2015. Autopsy. Retrieved from http://www.sleuthkit.org/autopsy/

[9] Fei-Fei, L., Fergus, R., and Perona, P. 2004. Learning generative visual models from few training examples: an incremental Bayesian approach tested on 101 object categories. IEEE. *CVPR 2004, Workshop on Generative-Model Based Vision.*

[10] NIST. 2006. Common Evaluation Measures (Appendix). *Proceedings of the Fifteenth Text REtrieval Conference (TREC 2006).*

[11] Jain, A. K., Klare, B., and Park, U. 2012. Face Matching and Retrieval in Forensics Applications. *IEEE Multimedia 19(1),* 20.