# Optimization of automatic nudity detection in high-resolution images with the use of NuDetective Forensic Tool

Pedro Monteiro da Silva Eleuterio and Mateus de Castro Polastro

***Abstract*—Due to amendments in Brazilian law, the possession of files containing child and teen pornography began to be crime and the need to detect this type of content at crime scenes increased. With this objective, the NuDetective Forensic Tool has been assisting forensic examiners to conduct such analysis in a timely manner. However, with the evolution of digital cameras and camcorders, image files are being produced with higher resolutions, which may result in an undesirable delay of processing these files. This work presents an approach to optimize the automatic nudity detection provided by NuDetective Forensic Tool by reducing the images resolution. Several evaluation experiments were performed and the results showed that it is possible to reduce the runtime in almost 90%, without significant changes in the nudity detection results.**

***Keywords*—child pornography, nudity detection, image processing, forensic computing.**

## 1. Introduction

The production and propagation of illegal digital multimedia content, especially images and videos involving child and teen pornography, have been increasingly common in crimes against children. Combined with the fast evolution of computers and the popularization of the Internet, the production and subsequent exchange of illegal files between people dispersed throughout the world occur more frequently, mainly through the World Wide Web (WWW), the peer-to-peer networks [1], e-mail, and instant messaging services.

These facts encouraged the natural evolution of the Brazilian legislation. Until November 25th, 2008, Brazilian law only criminalized the disclosure of such material, under Article 241 of the *Child and Adolescent Law* [2, 3]. After that date, with the publication of *Law 11,829*, possession of files of this nature also became a crime, with the inclusion of Article 241-B [4], which says:

> *"Acquire,* ***possess or store****, by any means, photo, video or other record that contains the scene of explicit sex or pornography involving children or adolescents" [4].*

Therefore, requests for forensic analysis in computer storage media such as hard disk drives, pen drives, optical media, and memory cards in order to verify the presence and distribution of child pornography are becoming more frequent. The experience of the authors shows that the greatest difficulty in this type of analysis is to identify the files containing child and teen pornography among the hundreds of thousands of files possibly existing in these devices. A recent study [5] showed that 148 (one hundred and forty-eight) files containing sexual abuse of a teenage girl were found after analysis of more than 300,000 image files and 1,100 video files. This exemplifies how hard is to find such files, because the analysis of images and videos is usually done visually.

Besides the need to detect files containing child and teen pornography in seized devices, the amendments in Brazilian Law has expanded the need to detect this type of material still in the crime scenes. Thus, forensic examiners need to identify these files in a timely way in order to avoid unnecessary seize of digital devices for later analysis in the laboratory, which often takes many days or even weeks to be done, losing the opportunity to arrest the suspect immediately. With this objective, the NuDetective Forensic Tool [6] has been assisting forensic examiners to conduct such analysis in a timely manner at the crime scene. The Tool performs pixel and shape analysis to automatic detect human nudity files (*Image Analysis*) and searches for file names that may contain typical expressions of child and teen pornography (*FileName Analysis*).

However, the continued evolution of digital multimedia devices such as cameras and camcorders, allowed the production of multimedia content with higher resolutions. Nowadays, digital cameras marketed in Brazil have an average resolution of 12 Megapixels. Many digital cameras may still exceed 20 Megapixels of image resolution. Therefore, the manipulation of these files in an automated way requires a computational effort that may delay the analysis of forensic examiners at the crime scene. This work has the main purpose to optimize the automatic nudity detection with the use of NuDetective Forensic Tool, reducing the runtime to process high-resolution images.

Another technique to search for illegal files widely used in forensic computing is to compare the hash value of the files. The hash is a value calculated from a one-way mathematical function [7] and serves as a unique identifier of the file. This approach calculates the hash value of all files on the suspicious material, comparing them with a list of hash values of known files, which were previously identified. Commercial tools such *Access Data Forensic Toolkit* [8] and *Guidance Encase* [9] have this functionality. Such technique would also be very useful to NuDetective Forensic Tool. Despite being an efficient method to identify known files, to perform the calculation of the hash value of thousands of files that can be stored on a hard disk drive, for example, may be a lengthy operation to be performed at the crime scene. Moreover, the success of this method depends on how complete is the list of hash values of known files. Therefore, new and unknown files, which could be produced by pedophiles, for example, would never be identified by this technique. However, depending on the operational viability and the time required, using this method (*Hash Analysis*) in conjunction with *Image Analysis* and *FileName Analysis* provided by NuDetective can make the results even more reliable and effective. Therefore, the implementation of *Hash Analysis* is also one of the purposes of this work.

In this scenario, the authors have studied ways to improve the NuDetective Forensic Tool to assist the forensic examiner to spend the smallest time as possible to detect child and teen pornography, especially in cases involving high-resolution images.

This paper is organized as follows: Section 2 presents the state of the art of pixel analysis and human nudity detection algorithms. A brief description of the NuDetective Forensic Tool, including the nudity detection algorithms used and its basic features, is present in Section 3. Section 4 presents the proposed approach for optimization of automatic nudity detection in high-resolution digital images, including several evaluation experiments. Section 5 briefly describes the new version 2.0 of the NuDetective Forensic Tool, including new features such as *Hash Analysis*. Sections 6 and 7 present the conclusions and future work, respectively.

## 2. Backgrounds

The detection of human nudity in digital images is mainly based on the identification of pixels that contain skin color and the way which the identified skin regions are related. To identify the skin color is important choose the color space and the parameters that will be used to define the skin color. On nudity identification, a comparison with human forms and the relationship between identified skin regions in the image are the main approaches to determine if an image contains or not human nudity.

Concepts such as color space, the definition of skin color and nudity detection are discussed below.

### 2.1. Color Space

A digital image can be described as a point matrix where each point (*pixel*) represents a color. Like any mathematical representation of a physical phenomenon, the color representation can be done in several ways, each containing its advantages and disadvantages [10]. Thus, the choice of color space will depend on the application type.

The RGB color space was originated from the old CRT (Cathodic Ray Tube). It defines the color representation by combining three primary colors: red (R), green (G), and blue (B) [11]. The scale of each of these components varies from 0 to 255. This is the most common color space used for storing digital image representation [12]. In RGB, brightness and color are coupled and thus not suitable for color segmentation in images with unknown light conditions [13]. To reduce the effect of luminance on the color representation, it is possible to use the normalized RGB, which consists of a transformation from RGB through a normalization process, shown in (1) [11].

$$r = \frac{R}{R+G+B}, g = \frac{G}{R+G+B}, b = \frac{B}{R+G+B} \quad (1)$$

Color spaces based on *Hue* and *Saturation* were proposed to allow an intuitive description of colors [11]. *Hue* defines the dominating color and *Saturation* defines the color based on brightness. These color spaces are known as HSI, HSV and HSL. The letters H and S represent *Hue* and *Saturation*, respectively, and the letters I, V and L, *Intensity*, *Value,* and *Lightness*, respectively, and are related to the luminance of the color.

Orthogonal color spaces such as YCbCr, YIQ, YUV, and YES, try to reduce redundancy in RGB color channels and represent colors by using components as independently as possible [12].

### 2.2. Skin Color Classification

The pixel classification of digital images, in order to determine skin regions, tries to identify each pixel as skin or not. There are many difficulties to classify skin pixels, because several factors affects its color, including: variable ambient light, the use of different digital cameras, which can produce different pixel colors of the same person under the same lighting condition, and; skin color varies from person to person [14]. In addition, several objects in the real world such as wood, leather, hair, and sand, among many others, have similar color to the skin. This often confuses a skin classifier in relation to the background of the image [15].

The human skin color, which mainly depends on the concentration of hemoglobin, melanin, and the lighting condition [16], is not located randomly within a color space. The skin colors are grouped within small areas of the color space, but these areas depend on the used color space [15].

Thus, many studies have been conducted to determine ways of classifying skin color pixels among existing color spaces. A research [17] has proposed a method for detecting skin color based on RGB color space. This is a method with simple rules that allow the quick construction of classifiers [11]. A basic arithmetic formula can describe the relationship between the three components of RGB color, as shown in relationship (2) below.

$$R > 95 \text{ and } G > 40 \text{ and } B > 20 \text{ and } \max\{R,G,B\} - \min\{R,G,B\} > 15 \text{ and}$$
$$|R-G| > 15 \text{ and } R > G \text{ and } R > B \quad (2)$$

Other color spaces have also been used to define a classifier of skin pixels. In [18], areas of color space that belong to the skin coloring set were obtained using the normalized RGB[1] and HSV color spaces from relations (3) and (4) below.

$$0.36 \leq r \leq 0.465,\ \ 0.28 \leq g \leq 0.363 \quad (3)$$

$$0 \leq H \leq 50, 0.20 \leq S \leq 0.68, 0.35 \leq V \leq 1.0 \quad (4)$$

The YCbCr color space was used in [19] to segment the regions of skin color through the chrominance information, where they achieved high rates of detection in images with good lighting conditions.

In addition to the detection of skin color by making rules, such as those cited above, there are probabilistic methods that are used to estimate the distribution of skin color without the use of a color space, but these methods use several images to train the classifier [11]. The result from this practice is known as *Skin Probability Map*, or PMS. This type of classifier, while providing good results in the detection of skin color and is independent of the color space used, require a lot of storage space to be fitted and representative images to train the model [11]. They are classified as *Skin Nonparametric Distribution Modeling*. An alternative for this type of classifier are the *Skin Distribution of Parametric Modeling*. Methods of this group are able to achieve high representation through the least number of training images, and use less storage space [12]. However, in some methods, such as *Mixture of Gaussians*, the performance depends strongly on the shape of the skin colors distribution and can be extremely slow in both training and classification processes [11].

### 2.3. *Nudity Detection*

The first step in the nudity detection in images is the identification of skin pixels. Once identified these points, is necessary to determine whether the regions bounded by skin color correspond to shapes of the human body. There are known methods in the literature that use image features to classify the presence of nudity. Also, there are methods that compare the segmented images by skin color with a database of pre-classified images.

One research [20] presents a number of ways to accomplish the nudity detection by extracting image features. It is possible to compare the size of the largest skin region found with the image size. The amount of identified skin areas is another important feature. Small skin regions in large quantities are not typical of nudity images. The percentage of skin in an image is also an indicative of the presence of nudity [20].

Another study [21] was also based on extracting image features to determine nudity. The algorithm described considers that nudity contains a lot of different skin tones and these regions are relatively close to each other. First, it calculates the percentage of skin pixels among total pixels of the image. Through experiments, was set a threshold higher than 15% so that the image continues to be evaluated. Otherwise, it defines that the image has no nudity. Then, the continuity of the three major identified regions of skin color is used in various calculations through comparison with thresholds empirically obtained.

Some other methods use an image database to train the classifier. One method [22] uses *Support Vector Machine* (SVM), which has learning skills in the image detection of human nudity. In the proposed model, is first verified the correlation between regions containing skin color and those that do not contain, to train the SVM. Then, the classifier uses this information to determine whether a new image contains nudity or not, based on information acquired during training. Another method [23] compares a suspect image with a set of features obtained from the analysis of a training database.

## 3. The NuDetective Forensic Tool

This section provides a quick review of the main features of the NuDetective Forensic Tool, as described in [6], to illustrate its basic operation, including its pixel analysis and its human nudity detection algorithms.

The automatic detection algorithms implemented by NuDetective was based on two main principles: high processing speed and reduced number of *false negatives*. Thus, both the choice of color space and identification parameters of skin color met these essential requirements, as described below.

### 3.1. *Color Space and Skin Color*

The NuDetective uses the RGB color space for representing digital color space, due to the ease and speed in color processing. Although the use of this color space does not allow the separation of luminance and color components, this separation in the skin color classification process does not necessarily lead to better results [11]. Moreover, one research [14] performs a formal proof that there is an optimum skin color classifier for any color space, i.e., the ability to detect skin color is independent of the color space used.

The calculation suggested by [17], described on the relationship (2), in Section 2, was used by NuDetective to relate the three components of RGB color space. The strategy to classify each pixel as skin color or not only applies to color images, which does not mean a significant disadvantage, because the vast majority of digital human nudity images, including child and teen pornography, is colored.

### 3.2. *Nudity Detection Algorithm*

For nudity detection, the Tool used the algorithm described in [21], which reached great results, as described in Section 3.4. Furthermore, the use of this algorithm met the proposed requirements of NuDetective, because it is computationally fast. Also, it does not require one previous database to classify

1 The relation (3) does not present a value for b, because on the normalized RGB, the sum of the components r, g and b is always equal to 1. Therefore, the value of b can be omitted [11].

the images. This is possible because this algorithm works with percentage of skin color and computational geometry to identify nudity in digital images [21]. The choice of this algorithm allowed NuDetective to run without a large computing infrastructure, which is desirable at crime scenes.

### 3.3. Basic Features

Developed with the *Java Standard Edition* (JSE) [24], the NuDetective Forensic Tool performs the automatic human nudity detection in images according to the algorithms described above, and many other options that can be configured by the forensic examiner to assist in the detection of child pornographic files.

First, the forensic examiner chooses *locals* to search, such as folders and/or disk partitions, and configures search options. Thus, the NuDetective scans all files of selected *locals* searching for images, submitting them to the nudity analysis process, called *Image Analysis*. In addition, the Tool can still check the names of all files or only image names, comparing them with a list of pre-defined keywords[2] that contains names and phrases commonly used to share child pornographic files over the Internet.

Several options can be configured by the forensic examiner, such as the image types to be searched, the method to search for images (by file extensions or by file signatures), filters by image sizes, how to compare names on *File Name Analysis*, among others. The Tool also allows the analysis of compressed files in ZIP format.

The NuDetective was implemented using Java *threads*, allowing the execution of some tasks in parallel. One of the developed *threads* is responsible for searching the selected *locals*, the second cares only about the nudity images processing and a third *thread* is responsible for displaying the results, feeding the results GUI in real time. So the forensic examiner can start to analyze the results even before the Tool finishes searching all files of the selected *locals*.

### 3.4. Previous Results

As described in [6], two evaluation experiments were carried out to measure the nudity detection rate of the Tool. In the first experiment, one controlled database was created with several images obtained at imageafter.com [25] and with nudity images obtained from random access of Internet sites, including human nudity of Asian, Black, and Caucasian people. The second experiment used a real hard disk drive seized during an operation to combat pedophilia in Brazil. The hard disk drive (HDD) was placed in the *Logicube Forensic Talon* [26], which was connected to one laptop's USB port, as shown in Figure 1. Thus, it ensured the preservation of all data in the hard disk drive, since the *Talon* acts as a write blocker of the storage device. This configuration is widely used at crime scenes by forensic examiners.

2 The list of keywords was defined considering the professional experience of the authors.

The results of the first experiment proved the ability of the Tool to automatically recognize suspicious nudity files. The results of the second experiment proved that the use of NuDetective is feasible and significantly reduces the time spent by forensic examiners to detect illegal content, with a high degree of reliability. The Tool took less than 13 minutes to search and analyze all files on the evidence drive, which had over more than three hundred thousand files stored. The results of experiments showed nudity detection rates around 95%, with low rates of *false positives*, combined with fast processing [6].

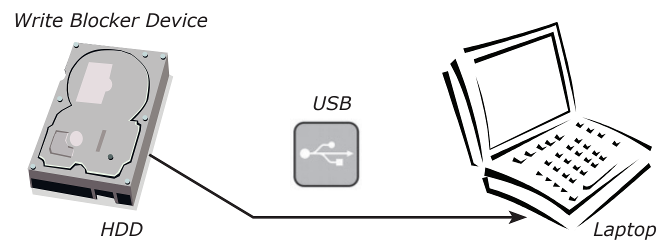


Fig. 1. Configuration widely used by forensic examiners: one write-protected hard disk drive connected to one laptop's USB port, through a write blocker device.

## 4. Optimization of Automatic Nudity Detection

The authors observed that the runtime of automatic nudity detection involving high-resolution digital photos may be high, as described in Section 4.2. Therefore, search for child pornographic files using the Tool on a hard disk drive containing many high-resolution images could take more time than desirable to be performed at a crime scene. This section describes our optimization approach, the evaluation experiments and results.

### 4.1. Material

Several experiments were performed in this work. For all of them, we used a laptop *Lenovo ThinkPad T60*, Intel Centrino 2.33 GHz, 2 GB of RAM, display adapter ATI Mobility Radeon X1400, running Windows XP SP3. Since the performed experiments tried to measure runtimes, the authors warn that the results are directly dependent on the hardware configuration used.

### 4.2. Runtime of Automatic Nudity Detection in High-Resolution Images

To calculate the runtime of automatic nudity detection in high-resolution images with the use of NuDetective, the authors performed experiments with digital photos of different resolutions. Two hundred random photos with resolution of 12.0 Megapixels were used in the experiments, some containing human nudity.

Later, using the software *Multiple Image Resizer .NET* [27], these pictures had their resolutions reduced to approximately 8.0, 4.0, 2.0, 0.7, and 0.075 Megapixels. Thus, we created a total of six sets of images containing 200 (two hundred) photos each.

These sets of images were copied to six different folders and submitted, one by one, to the *Image Analysis* provided by NuDetective in order to calculate the total runtime. The *FileName Analysis* was disabled. Table 1 shows the results for each of these sets, including the average runtime by an individual photo of each set. As expected, the runtime was directly proportional to the image resolution. This experiment showed that one high-resolution image took more than five seconds to be processed, which may be undesirable at crime scenes. However, NuDetective can quickly analyze images with medium and low resolutions – these images are the most shared through the Internet. All the times in Table 1 includes the I/O operation to read the photos from the hard disk drive.

Table 1. Runtime of *Image Analysis* to process the created six sets of 200 photos each.

| Resolution (Mpixels) | Total Time | Average Time for each Photo |
|---|---|---|
| 12.0 | 17m 21s | 5.20s |
| 8.0 | 12m 46s | 3.83s |
| 4.0 | 5m 48s | 1.74s |
| 2.0 | 3m 02s | 0.91s |
| 0.7 | 1m 09s | 0.34s |
| 0.075 | 0m 09s | 0.04s |

### *4.3. Proposed Optimization Approach*

Once proven the runtime variation of nudity detection according to image resolutions, the authors performed studies to reduce the runtime of *Image Analysis* provided by NuDetective Forensic Tool.

Our optimization approach is to reduce the images resolution before they were submitted to the nudity detection algorithms in an attempt to reduce the runtime of NuDetective. This approach has three major challenges that need to be solved in this work:

(i) Develop an algorithm to reduce the images resolution. It should be fast enough to not include undesirable delay, compensating negatively the time gained by reducing the images resolution;
(ii) Verify how similar will be the results for nudity detection of images with original and reduced resolutions, and;
(iii) Establish an optimal minimal image resolution to be used by the NuDetective.

### *4.4. Implementation of Proposed Optimization*

To achieve the main purposes of this work, the major three challenges described before must be solved. Each one of them is described below, including the development process and the performed experiments.

#### *4.4.1. Reduce the Images Resolution*

An algorithm to reduce the images resolution was developed by the authors. Its implementation was based on the use of Java classes `BufferedImage` and `Graphics2D`, as shown in Figure 2. The image with original resolution is represented by `originalImage` and the resulting image with reduced resolution is represented by `reducedImage`.

```
BufferedImage reducedImage = new
    BufferedImage (newImageWidth,
    newImageHeight, originalImage.
    getType());
Graphics2D graphics = img.
    createGraphics();
graphics.setRenderingHint(RenderingHints.
    KEY_INTERPOLATION, RenderingHints.
    VALUE_INTERPOLATION_BILINEAR);
graphics.drawImage(originalImage, 0,
    0, newImageWidth, newImageHeight,
    0, 0, originalImageWidth,
    originalImageHeight, null);
graphics.dispose();
```

Fig. 2. Main lines of Java code of the developed algorithm to reduce the images resolution.

Experiments with the algorithm to reduce images resolution were performed to calculate the runtime of this procedure. The results showed that, from an image already loaded into the memory of the Java program, the runtime ($t$) to reduce the photo resolution was less than a millisecond ($t < 0{,}001$s). Several images were used on the tests and most of them had resolution of 12 Megapixels. The results obtained with the reduction algorithm were satisfactory, since the runtime was very low and did not add significant delays in *Image Analysis* provided by the Tool.

#### *4.4.2. Nudity Detection Results for Different Image Resolutions*

Thus, to measure how similar will be the results for nudity detection of images with original and reduced resolutions, the authors performed some experiments. Since the result of *Image Analysis* for each picture is binary, varying between Nudity (positive) or Non-Nudity (negative), the authors used the six sets of 200 (two hundred) photos previously created to check the nudity detection results of the *Image Analysis*, which are shown in Table 2. The results of images with original resolution (12 Megapixels) were considered as a basis for comparison with the others sets.

Table 2. Results of the *Image Analysis* for each set of 200 photos.

| Resolution (Mpixels) | Positives (Nudity) | Negatives (Non-Nudity) | Variation (%) |
|---|---|---|---|
| 12.0 | 33 | 167 | (basis) |
| 8.0 | 33 | 167 | 0 % |
| 4.0 | 34 | 166 | 0.5 % |
| 2.0 | 34 | 166 | 0.5 % |
| 0.7 | 33 | 167 | 0 % |
| 0.075 | 35 | 165 | 1 % |

As shown in Table 2, the results of nudity detection algorithms used by NuDetective were very similar after examining the same pictures of different resolutions. The higher variation was only 1%. This result was satisfactory and allowed NuDetective to reduce the images resolution before submit them to *Image Analysis*, without significant changes in the detection results.

#### 4.4.3. Optimal Image Resolution

The authors made several other experiments to determine the minimum resolution that an image could be reduced in order to be analyzed by the nudity detection algorithms of NuDetective, without significant changes in results. After exhaustive tests and experiments with different resolutions smaller than 1 Megapixel, the authors concluded that the resolution of 0.01 Megapixel, or 10 Kilopixels, is the most appropriate in relation to the runtime and nudity detection results. Therefore, a new option was implemented in the Tool: enable or not the optimization of *Image Analysis*. When optimization is enabled, the algorithms of *Image Analysis* reduce the images resolution to a maximum height or width of 100 pixels, always keeping the aspect ratio. In this configuration, we obtained the best runtime with low variation in the results, which are in Table 3.

In this experiment, we used only images with original resolution (12 Megapixels) and they were reduced in real time by the new reduction algorithm provided by the Tool. This means that the total runtime includes the time to read the original photo files from the hard disk drive, which is an I/O operation, and naturally demands a great portion of time.

Table 3. Results of the optimization using the new version of NuDetective.

| Tasks | Optimization Disabled | Optimization Enabled |
|---|---|---|
| Total Runtime of 200 Photos with 12 Megapixels each | 17m 21s | 1m 52s |
| Average time for each Photo | 5.20s | 0.56s |
| Number of Positives (Nudity) / Negatives (Non-Nudity) | 67 / 133 | 65 / 135 |

### 4.5. Result Analysis

Our approach to reduce the image resolution to 0.01 Megapixels, by adjusting the height / width to a maximum of 100 pixels, showed very good results. Analyzing the results in Table 3, it is possible to check that the variation of the nudity detection results obtained by enabling the optimization was only 1% (2 different results in 200 photos). Therefore, the proposed optimization reduces the total runtime from 17 minutes and 21 seconds to 1 minute and 52 seconds, i.e., has achieved a reduction of approximately 90%.

The results obtained in different experiments with the proposed approach in this work showed that our optimization technique significantly reduces the runtime of NuDetective, making it even more feasible to be applied on real forensic cases to detect child and teen pornography at crime scenes.

The results also showed that the algorithms proposed by [17, 21] can process images of different resolutions, without significant changes in the nudity detection results. This variation can be explained by the fact that the process of reducing image resolution can change the skin regions originally detected, joining or segmenting those regions.

## 5. NuDetective 2.0: New Features

A new version 2.0 of NuDetective Forensic Tool was developed in this work. This section describes its new features and main improvements.

### 5.1. Hash Analysis Implementation and Evaluation

As described in Section 1 of this paper, calculate the hash value of files to search for previously known content is widely used in computer forensics. Thus, the new feature *Hash Analysis* was implemented in the new version of the NuDetective, allowing the forensic examiner to import a text file, which contains the list of known hash values, that will be compared with the files present in the selected *locals* (folders and/or disk partitions). The NuDetective also offers the option to calculate the hash values of only image/videos files, which is much more interesting, considering the main purpose to detect child and teen pornography. The forensic examiner can choose the hash type to be used: MD5, SHA-1, SHA-256 and SHA-512.

To measure the performance of *Hash Analysis*, one simple experiment was performed: we submitted our original 200 high-resolution images (12 Megapixels) to *Image Analysis* with optimization together with each type of *Hash Analysis* provided by the Tool. The main goal of this experiment is check the introduced delay of *Hash Analysis*. One text file with a list of a thousand hash values was used. We known, as showed in Table 3, the runtime of *Image Analysis* with optimization was 1 minute and 52 seconds. Table 4 shows the results of this hash experiment. As expected, the MD5 was the fastest. It introduces only 4 seconds delay in 200 high-resolution images and is the recommended hash type to use at crime scenes. Because NuDetective was implemented with Java *threads*, the delay introduced by *Hash Analysis* was minimal.

Table 4. Results of the hash experiment to verify the performance of Hash Analysis provided by NuDetective.

| Hash Type | Runtime of Optimized Image Analysis with Hash Analysis | Introduced Delay by Hash Analysis |
|---|---|---|
| MD5 | 1m 56s | 4s |
| SHA-1 | 2m 03s | 11s |
| SHA-256 | 2m 05s | 13s |
| SHA-512 | 2m 09s | 17s |

### 5.2. Other Features, New Graphics User Interfaces and More

The new version of NuDetective allows the forensic examiner to combine the three types of analysis (*Image*

*Analysis*, *FileName Analysis* and *Hash Analysis*), setting them according to the forensic needs of each case. It is possible to use all the three analysis type simultaneously, or choose from one or two specific types. Each analysis type has specific settings that can be changed by the forensic examiner.

Figure 3a illustrates the new main Graphical User Interface (GUI) of the Tool, which includes new options for enabling the new optimization of *Image Analysis* and also the *Hash Analysis*. It is possible to see that the graphical interface of the Tool is ready to support the analysis of video files (*Video Analysis* - still disabled in this version), which is under development by the authors, as discussed in future work (Section 7).

The table in the results GUI (Figure 3b) gained a new column indicating the *Hash Analysis* (HS), which is similar to the *Image Analysis* (Im) and *FileName Analysis* (FN). These columns indicate which analysis type classified the resulting suspect files.

The version 2.0 of the Forensic Tool NuDetective provides a lot of other fixes and internal improvements, including optimization in the source code to seek files, updating statistics and status messages, new control and error *logs*, and so on.

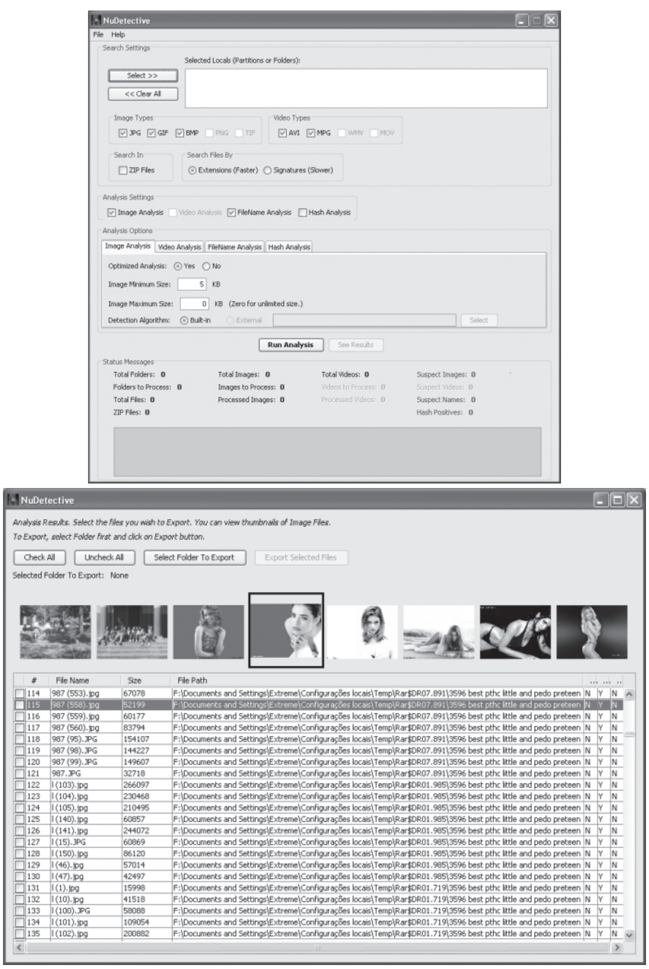


Fig. 3. (a) main GUI and (b) results GUI of NuDetective Forensic Tool.

## 6. Conclusions

One previous work [6] proved that the use of the NuDetective Forensic Tool to detect child and teen pornography was feasible. Its internal algorithms to detect human skin [17] and nudity [21] showed success detection rates of 95% with fast processing. However, when NuDetective tries to analyze high-resolution images, such as digital photos, one problem was found: it took more than 5 seconds for each 12 Megapixels photo, which is not desirable for a forensic examiner at a crime scene.

With the evolution of digital cameras and camcorders, anyone with such equipments can produce multimedia content with great quality and higher resolution. Nowadays, high-resolution photos are increasingly common on seized computers and storage devices to be examined by forensic experts. Therefore, this work presented an optimization technique to reduce the runtime of automatic nudity detection in high-resolution digital images with the use of NuDetective Forensic Tool.

The proposed approach reduces the images resolution to be analyzed by detection algorithms [17, 21] provided by the Tool. As results, besides providing a new version of the Tool, this work had solutions to the three main challenges described in Section 4:

- (i) An algorithm to reduce the images resolution was implemented in Java using `BufferedImage` and `Graphics2D` classes, and the time it takes to reduce a high-resolution image is less than a millisecond, which is a very satisfactory result;
- (ii) Our experiments showed that the results of automatic nudity detection provided by NuDetective had maximum variation of 1% between images with original and reduced resolutions. Particularly, this result allowed the proposed approach to be implemented without significant changes in results, and;
- (iii) The resolution of 0.01 Megapixels, or 10 Kilopixels, had a reduced runtime and good detection results, being chosen as the optimal resolution to reduce the images before submitting them to *Image Analysis* of NuDetective.

As expected, the performed experiments showed that the optimization proposed in this work is more effective as the images resolution increases. One experiment demonstrated a reduction of about 90% of the total runtime, from 17 minutes and 21 seconds to 1 minute and 52 seconds, analyzing two hundred high-resolution (12 Megapixels) photos.

This work also showed that the algorithms proposed by [17, 21] can process images of different resolutions, without significant changes in the nudity detection results.

The proposed optimization was implemented by the authors in the new version 2.0 of the NuDetective Forensic Tool. This version provides other new features, including a new type of file analysis: the *Hash Analysis*. As discussed, comparing the hash values of files can be quite useful in the detection of known files. Using this new version of the Tool, the forensic examiner can combine this new type of analysis (*Hash Analysis*) with *FileName Analysis* and *Image Analysis*, according to his needs, allowing greater confidence in the results of automatic detection of files containing child and teen pornography. Experiments showed that the introduced delay of *Hash Analysis* is minimal, because NuDetective is implemented with the use of Java *threads*.

The NuDetective Forensic Tool is free and available only to law enforcement. For instructions on how to get it, please contact the authors or send an e-mail to nudetective@gmail.com.

## 7. Future Work

Although the results of experiments had shown that the Tool is effective for use in real cases, NuDetective is still not able to analyze the contents of a typically file type used for nudity storage: the video. Thus, the implementation of nudity detection in videos through frame extraction is part of a project already started for further improvement in the software. As showed in Figure 3a, the new main Graphics User Interface of the Tool is ready to support this new feature.

Some other improvements that can be done in future already have been described in [6].

## Acknowledgements

The authors thank the Brazilian Federal Police, especially the Regional Oversight of Mato Grosso do Sul, for financial and logistic support.

## References

1. Saroiu, S. et. al. "A Measurement Study of Peer-to-Peer File Sharing Systems". Proceedings of the Multimedia Computing and Networking (MMCN). San Jose, Jan, 2002.
2. BRAZIL. 1990. Estatuto da Criança e do Adolescente. Lei número 8.069 de 13 de Julho de 1990. Brasil : s.n., 1990; [in portuguese].
3. BRAZIL. 2003. Lei número 10.764, de 12 de Novembro de 2003. Altera partes do Estatuto da Criança e do Adolescente. Brasil : s.n., 2003; [in portuguese].
4. BRAZIL. 2008. Lei número 11829 de 25 de novembro de 2008. Brasil : s.n., 2008; [in portuguese].
5. Eleuterio, P. M. S.; Machado, M. P. Identificação de autoria e materialidade em crimes de abuso sexual de criança/adolescente a partir da análise de arquivos multimídia. IV International Conference On Forensic Computer Science (ICoFCS 2009), Natal/RN, Brazil, 2009; [In portuguese].
6. Polastro, M. C.; Eleuterio, P. M. S. NuDetective: a Forensic Tool to Help Combat Child Pornography through Automatic Nudity Detection. 1st International Workshop on Visualization and Information Security Management (VISM'10). Bilbao, Spain. [To be published: August 2010]
7. Menezes, A. et. al. "Handbook of Applied Cryptography". CRC Press, 1996.
8. Access Data Forensic ToolKit (FTK). Available from http://www.accessdata.com. [Visited Jun, 2010].
9. Guidance EnCase Forensic. Available from http://www.guidancesoftware.com/computer-forensics-ediscovery-software-digital-evidence.htm. [Visited Jun, 2010].
10. Pascale, D. A Review of RGB Color Spaces … from xyY to R'G'B'. The BabelColor Company. Montreal, Canada, 2003.
11. Vezhnevets, V.; Sazonov, V.; Andreeva, A. A Survey on Pixel-Based Skin Color Detection Techniques. Proceedings Graphicon, 2003.
12. Kakumanu, P.; Makrogiannis, P.; Bourbakis, N. A survey of skin-color modeling and detection methods. Pattern Recognition 40, 2007.
13. Wong, K.; Lam, K.; Siu, W.. A robust scheme for live detection of human faces in color images. Signal Processing: Image Communication 18, 2003.
14. Albiol, A.; Torres, L.; Delp, E. J. Optimum color spaces for skin detection. International Conference on Image Processing, volume 1, pages 122–124, 2001.
15. Elgammal, A.; Muang, C.; Hu, D. Skin Detection - a Short Tutorial. Encyclopedia of Biometrics, Verlag, 2009.
16. Jones, M. J., Rehg, J. M. Statistical color models with application to skin detection. International. Journal of Computer Vision (IJCV), pages 81–96, 2002.
17. Kovac, J.; Peer, P.; Solina, F. Human skin colour clustering for face detection. International Conference on Computer as a Tool EUROCON 2003, Eds. B. Zajc, Ljubljana, Slovenia, September 2003.
18. Wang, Y.; Yuan, B. A novel approach for human face detection from color images under complex background. Pattern Recognition 34, pages 1983–1992, 2001.
19. Wang, H.; Chang, S. A highly efficient system for automatic face region detection in MPEG video, IEEE Trans. Circuits Systems Video Technol., pages 615-628, 1997.
20. Carlsson, A.; Eriksson, A.; Isik, M. Automatic detection of images containing nudity. Master thesis in intelligent systems Design, University of Gothenburg, Sweden, 2008.
21. Ap-apid, R. An Algorithm for Nudity Detection. Proceedings of the 5th Philippine Computing Science Congress, Cebu City, 2005.
22. Lin, Y.; Tseng, H.; Fuh, C. Pornography Detection Using Support Vector Machine. 16th IPPR Conference on Computer Vision, Graphics and Image Processing (CVGIP). Kinmen, ROC, 2003.
23. Wang, J.; Li, J.; Wiederhold, G.; Firschein, O. System for Screening Objectionable Images Using Daubechies' Wavelets and Color Histograms. Proc. of the International Workshop on Interactive Distributed Multimedia Systems and Telecommunication Services, pages 10–30, 1997.
24. Java SE Overview. Available from http://java.sun.com/javase/; [Visited Jun, 2010].
25. Image After. Available from http://www.imageafter.com; [Visited Jun, 2010].
26. Logicube Forensic Talon, Available from http://www.logicubeforensics.com/products/hd_duplication/talon.asp; [Visited Jun, 2010].
27. Multiple Imager Risizer .NET. Available from http://www.multipleimageresizer.net/; [Visited Jun, 2010].

**Pedro Monteiro da Silva Eleuterio and Mateus de Castro Polastro** Brazilian Federal Police, e-mails: {pedro.pmse, polastro.mcp} @dpf.gov.br, Campo Grande/MS - Brazil