Detecting and Preventing the Electronic Transmission of Illicit Images and Its Network Performance

Amin Ibrahim and Miguel Vargas Martin

University of Ontario Institute of Technology
2000 Simcoe Street North, Oshawa, ON, Canada
{amin.ibrahim,miguel.vargas-martin}@uoit.ca

Abstract. Child exploitation through the use of the Internet as a delivery and exchange tool is a growing method of abuse towards children. It is shown that a Stochastic Learning Weak Estimator learning algorithm and a Maximum Likelihood Estimator learning algorithm can be applied against Linear Classifiers to identify and filter illicit pornographic images. In this paper, these two learning algorithms were combined with distance algorithms such as the Non-negative Vector Similarity Coefficient-based Distance algorithm, Euclidian Distance, and a Weighted Euclidian Distance algorithm. Experimental results showed that classification accuracies and the network overhead did have a significant effect on routing devices.

Keywords: Computer forensics; network monitoring; image classification; feature extraction; pattern recognition; Netfilter, network overhead.

1 Introduction

The sexual exploitation of children remains a very serious problem and is rapidly increasing globally through the use of the Internet. Due to the borderless nature of the Internet, law enforcement agencies are continually challenged by the burden of painstakingly ascertaining, tracking, identifying, and capturing child pornographers. As the ratio of images being produced and distributed to the availability of man power to deter this issue increase, it is apparent that computerized assistance must be part of the solution.

The intense competition among the various digital equipment manufacturers in recent years has yielded lower cost digital imaging devices to enter low to medium income homes and has revolutionized the photography market. Criminals use readily available and affordable digital imaging equipment to record images of child abuse and these devices have greatly simplified the production and distribution of illicit images.

Due to the advancement of technology, the problem of child pornography continues to grow rapidly, and a single instance to reflect the severity of this problem is a 2006 report by Internet filter review, shown in Table 1. An alarming fact from this same report sates the daily child pornography requests made to the Gnutella peer-to-peer file sharing network is about 116000[1].

S. Goel (Ed.): ICDF2C 2009, LNICST 31, pp. 139–150, 2010.
© Institute for Computer Sciences, Social-Informatics and Telecommunications Engineering 2010

Table 1. Internet pornography statistics in year 2006 [1]

Internet Pornography Statistics	
Pornographic pages	420 million
Daily pornographic search engine requests	68 million (25% of total search engine requests)
Daily pornographic emails	2.5 billion (8% of total emails)
Internet users who view porn	42.7%
Received unwanted exposure to sexual material	34%
Average daily pornographic emails/user	4.5 per Internet user
Monthly Pornographic downloads (Peer-to-peer)	1.5 billion (35% of all downloads)
Daily Gnutella "child pornography" requests	116,000
Websites offering illegal child pornography	100,000
Youths who received sexual solicitation	1 in 7

Though difficult, law enforcement agencies have played their role in combating the transmission of child pornography. In the past, law enforcement agencies have resorted to operating fraudulent websites which showcase an assortment of supposed child abuse images. These websites can then ask for registration and also for credit card payments to activate membership. Once an identity is established, police investigate the individual further. In 2003, various police agencies from Britain, North America and Australia successfully executed Operation Pin which involved the creation of fraudulent websites to attract pedophiles, and obtain the details of their identities [2].

In January of 2001, The United States Federal Bureau of Investigation commenced Operation Candyman by monitoring the electronic mail and chat room interactions of suspected pedophiles. Within several months, agents had assembled and identified a database of thousands of individuals who were then tracked to their homes and workplaces using IP addressing data obtained from service providers [3].

This paper is organized into the following sections: Section 2 discusses related works including the various image recognition algorithms, and other commercially available products to filter unwanted sexual content from networks. Section 3 discusses the design and implementation of the developed software through the various stages. Section 4 details experimental results such as system accuracy and system network performance. Section 5 is dedicated to the conclusion and potential direction for future work.

2 Related Work

Detection and prevention of electronically transmitted illicit pornography involves complex technical measures in relation to still images and moving video. Some notable works which are promising have been conducted and others are still being experimented with. As of yet, none offers a total solution, however, cannot be discounted as they are certainly a crucial element to the overall solution.

Currently available filtering tools are divided into three major groups: host-based, proxy-based and network-based. Host-based filtering allows users to install an application which merges with the operating system itself to provide protection, or install software which merges with a Primary Internet access application, such as a web browser. Examples of this type of filtering from commercially available products include NetNanny and Cyber Patrol.

Proxy-based filtering tools are usually employed in businesses to block inappropriate content and websites. This blocking mechanism can be used within a server relay used by employee workstations to provide internal Internet access. Workstations connected through this server filters websites that are previously blacklisted by addresses or keywords. Generally, because of load considerations, this particular type of filtering uses a static method.

Network-based filtering systems usually use packet classification by analyzing packet header data, and are focused towards intrusion detection and prevention systems, or Quality of Service (QoS) features. However, [8-10] proposes the use of Stochastic Weak Estimation coupled with linear classifiers as another payload inspection method, and this approach aims to derive linear classifications using statistical identifiers of IP packets.

Che-Jen et al. [4] proposed a digital feature retrieval mechanism for JPEG image files. The scheme extracts JPEG byte stream data at a packet level and it uses its DC coefficients to search for suspicious files. The problem with this approach is that it is prone to attacks such as scaling, transformation, cropping, lightening, darkening, noise corruption and network transition issues like packet loss, on JPEG files.

Whitehead et al. [5] implemented a method of classifying Internet objects using descriptor coefficients, such as name coefficient, text coefficient, image coefficient, audio coefficient, video coefficient, plug-in coefficient, and relational coefficient. The image data is analyzed to determine whether it contains adult content inside an Internet object using predefined skin tone ranges in Hue-Saturation-Value (HSV) color space. Forsyth [6] reported that human skin has Hue values between 0 to 25 (of a maximum 180) and saturation values between 50 and 230 (of a maximum 255). Whitehead used a statistical analysis of data settings to further refine the ranges, to Hue values between 2 to 18 and saturation values between 80 and 230.

Similarly, the WebGaurd [7] system intends to automatically detect and filter adult content from the Internet. WebGuard uses a crawler based system to extract relevant data, combines textual and image content, and the URL name of a site to construct its feature vector. To improve performance, an analysis using a skin color pixel mode is used.

Host-based and proxy-based solutions are effective for the traditional blockage of restricted content; each has very severe drawbacks and safety mechanisms which can be easily subverted. Host based filtering can be circumvented by simply uninstalling the software, or by installing and loading a different operating system on the same system. Proxy based solutions can be circumvented by using another proxy, or determining addresses for edge routing devices to connect through to an external network.

3 System Design and Implementation

The system is composed of three major blocks – image packet extraction and decoding, feature extraction and image classification. It is capable of working with both compressed and uncompressed images. To accomplish image classification at the network layer, traffic capture is required. Netfilter is used to capture each packet transiting the network. Netfilter is a subsystem in the Linux 2.4 kernel which facilitates packet filtering, network address translation (NAT) and connection tracking, through the use of hooks in the kernel network code.

Shown in Fig. 1 is the diagrammatic representations of the overall system – the first is for a compressed data stream (byte stream) and the latter for decoded RGB color space. The input to the system is the Internet traffic which consists of packets. As each packet is captured by the system the image filter is engaged and processes the packets that contain images. The assembled image packets then sent JPEG decoder which converts the image binary sequence to its RGB components. The feature extractor extracts information based on two algorithms, the Maximum Likelihood Estimator and the Stochastic Learning Weak Estimator. The obtained features are then compared against two feature vectors, which are obtained during the training stage of the system.

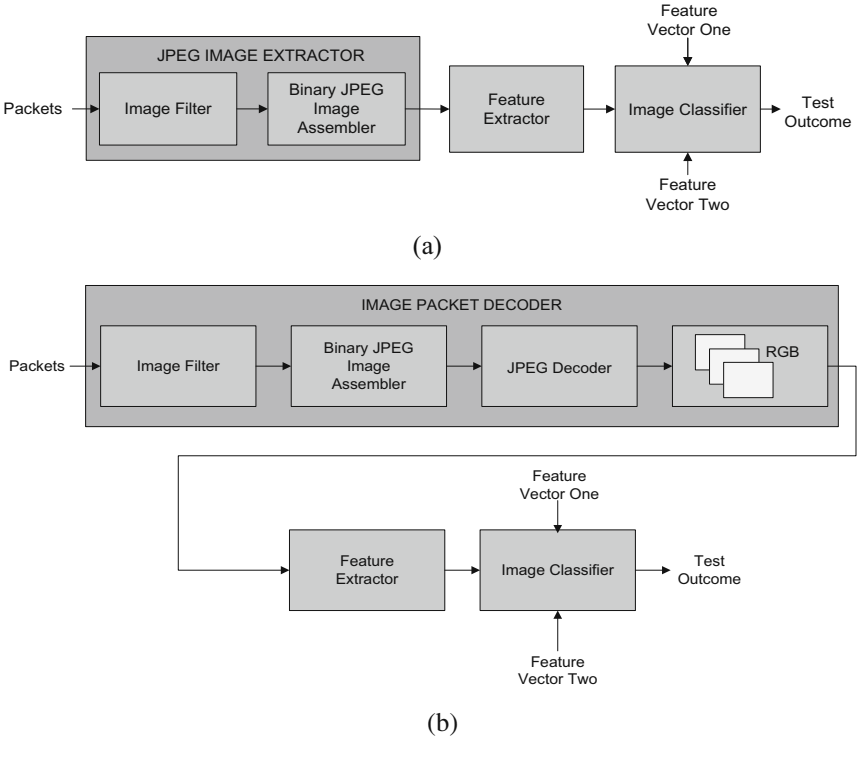

Fig. 1. Block diagram of the proposed method. (a) for byte stream data and (b) for RGB color space data.

3.1 Image Extraction

Our assumption is that every packet which passes through our system does so through a gateway on which our system is implemented. The system processes traffic by differentiating two classes of packets, JPEG image, or non-image packets. For our implementation we only consider one of the most popular and widely used image type, JPEG. The majority of images are fragmented into smaller pieces as per the Maximum Transmission Unit of most TCP/IP networks. To reconstruct these images in preparation for analysis, the system has three buffers that are successively used to store images as they transit the network. To extract only image packets from overall network data, a JPEG image marker, Start of Image (SOI), is used to identify image packets, which are then placed into an available buffer.

TCP sequence numbers are used to extract the remaining image data until an End of Image (EOI) marker is found within our payload. When the buffer is full, the payload is placed onto the JPEG decoder and the TCP header is processed for further analysis if needed. The JPEG decoder converts the compressed image data to its RGB color component. Note that in a byte stream method, the entire contents of the buffer data are directly passed onto the classification module.

3.2 Training

The purpose of the training stage is to extract feature vectors of two supervised classes: nude and non-nude images. This is done using the spectral characteristics and the byte streams of JPEG images. This training is accomplished using a Maximum Likelihood Estimator and a Stochastic Learning Weak Estimator.

3.2.1 Maximum Likelihood Estimator

The Maximum Likelihood Estimator (MLE) is one of the most popular statistical methods used for fitting a mathematical model to some data. The idea behind maximum likelihood estimation is to determine the parameters that maximize the probability (likelihood) of the sample data. Loosely speaking, the likelihood of a set of data is the probability of obtaining that particular set of data, given the chosen probability distribution model. For our system, we used 256 symbols for every components of our color space and also 256 symbols for byte stream.

The specific figure of 256 symbols was chosen because for any byte stream or RGB color space data, the range is defined from 0 to 255. Now, we want to estimate $S = [s_0, s_1,\ldots,s_{255}]$ and s_i calculated as:

$$s_i = \textit{frequency of symbol } i \textit{ / total symbols}. \tag{1}$$

Where, $\sum_{i=0}^{255} s_i = 1$.

For the color space samples, we have 256 color realizations (symbols) of every component. Thus, we want to estimate the outcome of each symbol for each color

component, namely red, green and blue. Therefore $S = \{S_R, S_G, S_B\}$ and S_R, S_G, and S_B are defined as $[s_{R0}, s_{R1},\ldots,s_{R255}]$, $[s_{G0}, s_{G1},\ldots,s_{G255}]$, and $[s_{B0}, s_{B1},\ldots,s_{B255}]$ respectively.

$$s_{Ci} = \textit{frequency of symbol } i \textit{ of color } C \textit{ / total number of pixels}. \tag{2}$$

Below is an explanation of MLE algorithm

```
Algorithm: MLE(Image,row,col)
for i = 0 to 255
   set V[i]=0
for i = 0 to row*col
   set V[Image[i]] = V[Image[i]]+1
for i = 0 to 255
   set V[i]= V[i]/(row*col)
```

3.2.2 Stochastic Learning Weak Estimator

The Stochastic Learning Weak Estimator, or SLWE, was proposed by Ommen et al. [11] as a replacement for the MLE algorithm's deficiencies. In particular, its proficiency in quickly capturing changes in the source of distribution for a particular set of data. For each estimate performed, each instant is updated, based on the values of the current sample.

Let V be a multinomial distributed random variable, which takes on the values from the set $\{1,\ldots,255\}$. We assume that V is governed by the distribution $S=[s_1,\ldots,s_{255}]^T$ where $V = i$ with probability s_i, and $\sum_{i=0}^{255} s_i = 1$.

For our model we let $V(n)$ be a realization of V at time n. Here the aim is to estimate s_i for $i = 0,\ldots,255$. We achieve this by maintaining a running estimate $V(n)=[p_0(n),\ldots,p_{255}(n)]^T$ of S, where $V_i(n)$ is the estimate of s_i at time n, for $i = 1,\ldots255$. Then, the value of $V_i(n)$ is updated as per the following simple rule:

$$V_i(n+1) = \begin{cases} V_i + (1-\lambda)\sum_{j\neq i} V_j, & V[n] = i \\ \lambda V_i, & V[n] \neq i \end{cases} \tag{3}$$

where $i = 0,\ldots,255$ and λ is training constant. [8 and 12] reported that the value of λ is approximately 0.999. For our experiment, we found that the optimal value of λ is 0.9995. Furthermore, [11] show that when $n\rightarrow\infty$, the expected value of $V_i(\infty)$ is equal to s_i and therefore $\sum_{i=0}^{255} V_i(\infty) = 1$. SLWE algorithm is shown below.

```
Algorithm: SLWE(Image,row,col,λ)
   for i = 0 to 255
      set V[i]= 1/256
   for i = 0 to row*col
      set temp = SLEW_sum(V,Image[i],λ)
```

```
        for j=0 to 255
            set V[j] = λ*V[j]
        set V[Image[i]]=temp
Algorithm: SLWE_sum(V,n,λ)
    set sum = 0
    for i = 0 to 255
        set sum = sum +V[i]
    set sum = sum - V[n]
    set sum = V[n] + (1-λ)*sum
```

3.2.3 Complexity of MLE and SLWE Algorithms

A significant concern of this research is network performance. Any large amounts of overhead introduced on the network would cause the system to be unusable in practical application. For this reason, it is very important to choose an algorithm that shows a best result for its accuracy and network performance. One way to analyse the network performance is to calculate the running time complexity of our algorithms. Form the above mentioned two estimators we can easily observe that both MLE and SLEW algorithms have a complexity of *O(n)*, where n is the number of bytes for compressed byte stream data or the number of pixels for RGB color space data. However, when comparing the number of operations used by these algorithms, we see that the SLWE algorithm use 255 more steps than that of MLE to update its feature vectors in every step.

3.3 Classification Distances

Statistical distance measure is defined as the distance between two probability distributions. It captures correlations or variations between attributes of feature vectors. Among all statistical distance classification functions, we chose and tested the three most widely-used in pattern recognition and three others that showed excellent performance in [13 and 14]. This list includes the Euclidean distance (ED), the Weighted Euclidian distance (WED), Variational Distance (VD), Counter Distance (CD), Cosine Distance (CosD) and Non-negative Similarity Coefficient-based distance (NVSC).

Table 2 shows the six classification distances and their formula for feature vectors with 256 symbols $V = (v_0,\ldots,v_{255})$ and $V' = (v_0',\ldots,v_{255}')$. For WED, σ_i is the standard deviation of v_i and defined as.

$$\sigma_i = \sqrt{\frac{1}{N}\sum_{i=0}^{N}(v_{ij} - \overline{V}_i)^2} \tag{4}$$

Where $\overline{V}_i$ is the mean of V_i.

Table 2. Classification Distances

Classification Distance	Formula
Euclidian Distance (ED)	$d(V,V') = \sqrt{\sum_{i=0}^{255}(v_i - v_i')^2}$
Weighted Euclidian Distance (WED)	$d(V,V') = \sqrt{\sum_{i=0}^{255}\frac{(v_i - v_i')^2}{\sigma_i^2}}$
Variational Distance (VD)	$d(V,V') = \sqrt{\sum_{i=0}^{255}\mid v_i - v_i' \mid}$
Counter Distance (CD)	$d_1 = Total\ count\ of \mid v_{1i} - v_i' \mid < \mid v_{2i} - v_i' \mid$ $d_2 = 256 - d_1$
Cosine Distance (CosD)	$d(V,V') = 1 - \frac{\sum_{i=0}^{255} v_i v_i'}{\sqrt{\sum_{i=0}^{255} v_i^2}\ \sqrt{\sum_{i=0}^{255} v_i'^2}}$
Non-negative Vector Similarity Coefficient-based Distance (NVSC)	$d(V,V') = \frac{\sum_{i=0}^{255}\min(v_i, v_i')}{\sum_{i=0}^{255}\max(v_i, v_i')}$

4 Experimental Results

The experiments for this work were conducted with three host computers and a router. The three hosts were connected to a Linux-based (Fedora Core 6) system where each host exchanged data passing through the Linux server (router). The Linux router used in this experiment was executed on an Intel x-86 based Pentium 4 CPU, with a clock frequency of 2.4GHz. The hosts for the experiment were comprised of three physically separate systems with Intel x-86 based Pentium 4 processors.

The experimental image database was comprised of a total of 3000 random non-child pornography images from the Internet. Most of the test images were sampled from Google images, Yahoo! images and some custom made images. Due to time constraints and different circumstances, all tests were done on legal images. Further tests on child pornography images with the help of Toronto Police Service, Sex Crimes Unit will be done in near future. Our experiment was divided into three categories: image classification, image matching and network performance. For our network performance, Wireshark was used to measure the delay of transmitting files with filter and without filter engaged.

4.1 Image Classification Performance

The classification stage is divided into two stages – training and classification, and is done for both compressed binary data and RGB data. All images used in our experiments were legal. No actual child pornography images were ever collected, seen, sought, or used.

Table 3. Success rate for nude images using compressed byte stream image data

Algorithm	Success rate					
	ED	WED	VD	CD	CosD	NVSC
MLE	63.6	65.4	68.3	61.7	66.5	67.9
SLWE	67.5	66.4	69.4	60.3	64.3	69.1

Pre-classified images were separated into two categories, nude and non-nude. For each category of images, we preselected 150 images of either fully nude content, or, no nudity content, with an average size of 640×480 pixels, (~75kB). Two methods of training were utilized for our work; compressed byte stream and RGB color space.

In the classification stage, the system was engaged to intercept packets for all network traffic that passed through our network. This was then tested with 3000 images for classification using six distance algorithms as explained in section 3. Table 3 shows the success rate for byte stream data and VD was optimal for both MLE and SLWE algorithms. Also, we observed that the success rate for NVSC was not far away from that of VD.

Table 4. Best performances by each classification distance algorithms and their respective scan area percentage for height scan and radius scan

Algorithm	Distance algorithm	Height Scan		Radius Scan	
		Best performance	Corresponding scan area	Best performance	Corresponding scan area
MLE	ED	68%	100%	70%	90%
	WED	72%	80%	70%	90%
	VD	65%	90%	68%	80%
	CD	67%	70%	68%	90%
	CosD	70%	70%	75%	90%
	NVSC	74%	70%	76%	80%
SLWE	ED	71%	80%	74%	80%
	WED	76%	80%	76%	80%
	VD	74%	80%	74%	80%
	CD	67%	80%	69%	70%
	CosD	75%	80%	76%	80%
	NVSC	75%	70%	77%	80%

Image classification for RGB color space is done by using a "height" scan and a "radius" scan. The height scan and radius scan method work by beginning with a full image scan while noting accuracy, and then gradually reducing the processed image areas to 20%. The radius scan similar to height scan but the scanning process is done in a circular manner where we start from 20% of the image from the middle and gradually increase it to 100%. Height scan and radius scans serve two primary reasons for their use. First, since most images which enter our network are fragmented according to the MTU, usually 1500 bytes, the accuracy of our system may be determined from analysis of only a portion of the image. Second, it was observed that system performance will increase if features which are determined to have no effect on image analysis are removed.

Table 4 summarizes the overall best performance by each classification distance algorithms and their respective scan area when using RGB color space data. From all these methods SLWE algorithm coupled with NVSC achieved the best performance. Also, it was observed that the optimal to scan was 75-85% of the area of the test image. Overall, the SLWE algorithm improved the system performance by ~2 – 4%. Other important observation was that when the individual color components were compared, the success rate was not distant from its RGB counterpart.

4.2 Network Performance

Using Wireshark in capture mode, network performance was measured while the first 100 images were transferred between two hosts. The average size of these images was 640×480 pixels, (~75kB). A time measurement was performed between filtered and unfiltered processing to determine the amount of overhead generated by the system.

Table 5. Network performances for compressed byte stream, RGB color space and Blue component of the image; all results are in seconds

Algorithm	Distance algorithm	Byte stream	RGB	Blue
MLE	ED	17.20	24.70	20.30
	WED	18.20	25.30	21.90
	VD	17.40	24.75	20.80
	CD	17.10	24.90	20.98
	CosD	17.70	24.67	20.56
	NVSC	17.80	24.59	20.49
SLWE	ED	25.20	38.90	32.40
	WED	28.00	41.40	33.70
	VD	25.40	39.20	32.80
	CD	26.00	38.95	32.99
	CosD	24.60	39.30	32.60
	NVSC	24.80	39.50	32.80

The system was enabled in various test modes such as byte stream, RGB filtering mode and for one color component using MLE and SLWE. The time taken to transfer 100 images without the filter was 15.5 seconds.

As a result of these experiments, tests showed the best timings were achieved when using byte stream with MLE, 11% to 17% of overhead occurred when using ED and WED, respectively. However, the SLWE algorithm for byte stream data produced, network overhead as high as 80% to as low as 58%. A large factor for SLWE overhead is that it required 255 additional operations than that of MLE to update its feature vectors in every step. When we considered uncompressed image data the network overhead increase almost by 2 fold. Table 5 summarizes the processing times and the overhead when considering compressed byte stream, uncompressed RGB color space and one the B component of the image respectively.

5 Conclusion and Recommendation

The electronic transmission of child pornography remains a large problem and this research is imperative in its solution. Our approach utilized a SLWE algorithm coupled with a Linear Classifier and also used a MLE coupled with a linear classifier. Our experiments showed that the SLWE coupled with the NVSC distance algorithm was highly accurate for the RGB color spaces of images. This method proved to adversely affect network performance with its extra overhead and may be deemed unsuitable for network layer approaches. For byte stream image data, an MLE and VD based algorithm method proved the best in network layer performance with the lowest overhead, but accuracy was 4 to 5 percent lower than the SLWE coupled with the NVSC distance algorithm method. MLE coupled with and NVSC distance algorithm showed nearly the same results.

Network performance for this research was a very important issue. Any large amounts of overhead introduced on the network would render the system unusable in any practical application. Algorithms which showed promising results in terms of accuracy, such as SLWE coupled with WED or NVSC, are unlikely to be used in network-based implementations, since SLWE has roughly 3 to 4 times higher overhead than MLE.

It was also observed that when scanning 75 to 80 percent of a given test image using the radius scan method, a better result was achieved. This was determined to occur since extra processing time was avoided since 10 to 15 percent of surrounding areas within images were deducted.

Our system was disadvantaged since every image was classified, even when images obtained from certain known sources could have easily and efficiently been checked against a blacklist first. This could occur using a text-based method of analyzing image meta-data, URL information, keywords or filenames. The system described in this paper would have proved more efficient if those text-based scanning methods were used first, and then, our system would be triggered to scan further in the event that text-based methods passed inspection.

Acknowledgement

We would like to thank the Toronto Police Service, Sex Crimes Unit, for providing their feedback in some stages of this work. We also thank the continuing generous support of anonymous Canadian foundation requirements and expectations of this work. Your support was a key contribution (name withheld upon their request).

References

1. Internet Pornography Statistics, http://internet-filter-review.toptenreviews.com/internet-pornography-statistics.html
2. Warner, B.: Police to Launch International Cyber Child Porn Sting (2003), http://www.castlecops.com/a4498Police_to_Launch_International_Cyber_Child_Porn_Sting.html
3. Kevin, J.: FBI Arrests 40 in Child Porn Sting (2002), http://www.usatoday.com/tech/news/2002/03/18/net-porn.htm
4. Che-Jen, H., Wei-Cheng, L., Jung-Shian, L.: An Efficient Packet-level JPEG Forensic Data Collection. Future Generation Communication and Networking 2, 108–113 (2007)
5. Ryan, M.P., Whitehead, A.D.: Method and Device for Classifying Internet Objects and Objects Stored on Computer-readable Media. US patent document No: 7383282, Application No: 09/978,182, Application Date, October 17 (2001)
6. Forsyth, D., Fleck, M.: Automatic Detection of Human Nudes. International Journal of Computer Vision, 63–77 (August 1999)
7. Hammami, M., Chahir, Y., Chen, L.: WebGuard: Web Based Adult Content Detection and Filtering System. In: IEEE/WIC International Conference on Web Intelligence, October 13-17, vol. 2, pp. 574–578 (2003)
8. Shupo, A., Martin, M.V., Rueda, L., Bulkan, A., Chen, Y., Hung, P.C.K.: Toward Efficient Detection of Child Pornography in the Network Infrastructure. IADIS International Journal on Computer Science and Information Systems 1, 15–31 (2006)
9. Munish, C., Martin, M.V., Rueda, L., Hung, P.C.K.: Toward New Paradigms to Combating Internet Child Pornography. In: Canadian Conference on Electrical and Computer Engineering, CCECE 2006, May. 2006, pp. 1012–1015 (2006)
10. Chopra, M., Martin, M.V., Rueda, L., Hung, P.C.K.: A Source Address Reputation System to Combating Child Pornography at the Network Level. In: IADIS International Conference on Applied Computing (February 2006)
11. Oommen, B.J., Rueda, L.: Stochastic Learning-based Weak estimation of Multinomial Random Variables and its Applications to Pattern Recognition in Non-stationary Environments. Pattern Recognition 39, 328–341 (2006)
12. Oommen, B.J., Rueda, L.: On Utilizing Stochastic Learning Weak Estimators for Training and Classification of Patterns with Non-stationary Distributions. In: Proceeding of the 28th German Conference on Artificial Intelligence, Koblenz, Germany, pp. 107–120. Springer, Heidelberg (2005)
13. Xue, Y., Tong, C.S., Zhang, W.: Evaluation of Distance Measures for NMF-based Face Recognition. In: International Conference on Computational Intelligence and Security, November 2006, vol. 1, pp. 651–656 (2006)
14. Qian, G., Sural, S., Gu, Y.: Pramanik: Similarity Between Euclidean and Cosine Angle Distance for Nearest Neighbor Queries. In: Proceedings of the 2004 ACM Symposium on Applied Computing, March 2004, pp. 1232–1237. New York (2004)