DETECTING AND PREVENTING THE ELECTRONIC TRANSMISSION OF ILLICIT IMAGES

BY

AMIN ABDURAHMAN IBRAHIM

A THESIS SUBMITTED IN PARTIAL FULFILLMENT

OF THE REQUIREMENTS FOR THE DEGREE OF

MASTER OF APPLIED SCIENCE

IN

ELECTRICAL & COMPUTER ENGINEERING

THE FACULTY OF ENGINEERING AND APPLIED SCIENCE

UNIVERSITY OF ONTARIO INSTITUTE OF TECHNOLOGY

APRIL, 2009

CERTIFICATE OF APPROVAL

ABSTRACT

The sexual exploitation of children remains a very serious problem and is rapidly increasing globally through the use of the Internet. This work focuses on the current methods employed by criminals to generate and distribute child pornography, the methods used by law enforcement agencies to deter them, and the drawbacks of currently used methods, as well as the surrounding legal and privacy issues. A proven method to detect the transmission of illicit images at the network layer is presented within this paper. With this research, it is now possible to actively filter illicit pornographic images as they are transmitted over the network layer in real-time. 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 thesis, these two learning algorithms were combined with algorithms such as the Non-negative Vector Similarity Coefficient-based Distance algorithm, Euclidian Distance, and Weighted Euclidian Distance. Based upon this research, a prototype was developed using the abovementioned system, capable of performing classification on both compressed and uncompressed images. Experimental results showed that classification accuracies and the overhead of network-based approaches did have a significant effect on routing devices. All images used in our experiments were legal. No actual child pornography images were ever collected, seen, sought, or used.

DEDICATION

To God, who has made everything possible.

To my parents, your love and support for my education was an inspiration for me to strive.

To my brothers and sisters, I am in debt to your understanding and compassion.

To my wife, my loving companion, Inas Abdella, your comfort has guided me throughout this journey.

To all of my dear friends, for your patience and assistance.

To my colleagues, teachers, and professors, I thank-you for your wisdom and dedication.

"All that is necessary for evil to succeed is for good men to do nothing."

– Edmund Burke

ACKNOWLEGMENTS

I would like to sincerely thank and honour a range of people who were vital to the development and completion of this work. This publication would not have been possible without their kind assistance.

My supervisor, mentor and professor, Dr. Miguel Vargas Martin, for thoroughly and thoughtfully granting in-depth insight, advice and conclusive problem solving methods which allowed me to achieve the completion of this work. I appreciate all that you have helped me with and I look ahead to the future with you in mind.

Dr. Ali Grami, whose enlightening conversations and guidance served to provoke and encourage me to reach my goals with precision. My research would not have concluded without the direction you so humbly and unselfishly granted.

My colleagues at the University of Ontario, our inspiring conversations, your constant company and your ideas created a pleasant environment for me to work in and I could not have completed my work without your presence.

The Royal Canadian Mounted Police and the Metro Toronto Police Service, for providing the requirements and expectations of this work. Your support was a key contribution to the establishment of this thesis. I wish your forces the best and hope that I have helped to advance your missions.

TABLE OF CONTENTS

LIST OF FIGURES

LIST OF TABLES

NOMENCLATURE

Symbols	Meaning / Use
$s_{x,y}(x,y)$	Intensity of pixel at location row x, and column y
$S_{u,v}(u,v)$	Quantized DCT coefficient at location row u, column v
C_n	Quantization constant used by DCT
S	Short form of Intensity of pixel / the set probabilities of all multinomial random variables
s	Short form of DCT coefficient
s_i	Probability of symbol multinomial random variable i
D	An 8 × 8 constant formed by the DCT values and constants
$p_i(n)$	Probability of a symbol i at time n
$V_i(n)$	Realization of feature vector i at time n
v_i	Feature vector i
v_{ij}	Feature vector i for image j
λ	Learning constant used in SLWE algorithm
$d(V,V')$	Classification distance between two feature vectors
σ_i^2	Standard deviation of feature vector i
$\bar{V}_i$	Mean of feature vector i

ACRONYMS AND ABBREVIATIONS

BMP	Bitmap
CALEA	Communication assistance for law enforcement Act
CAM	Content Addressable Memory
CBIR	Content-Based Image Retrieval
CCTV	Closed Circuit Television
CD	Counter Distance
CETS	The Child Exploitation Tracking System
CosD	Cosine Distance
DC	Direct Current
DCT	Discrete Cosine Transform
ECS	Entropy Coded Segments
ED	Euclidian Distance
EOB	End of Block (JPEG marker)
EOI	End of Image (JPEG marker)
FDCT	Forward Discrete Cosine Transform
FEM	Finite Element method
FSD	Fourier Shape Descriptors
FIS	Fat Inverted Segment
HSV	Hue-Saturation Value
IDCT	Inverse Discrete Cosine Transform
IP	Internet Protocol

JPEG	Joint Photographic Experts Group
MCU	Minimal Coded Unit
MI	Moment Invariants
MLE	Maximum Likelihood Estimator
NAT	Network Address Translation
NIDS	Network Intrusion Detection Systems
NVSC	Non-negative Vector Similarity Coefficient-based distance
P2P	Peer-to-Peer
PIPEDA	Personal Information Protection and Electronic Document Act
QoS	Quality of Service
RAM	Random Accessible Memory
RGB	Red/Green/Blue
RST	Restart (JPEG marker)
SLWE	Stochastic Learning Weak Estimator
SOF	Start of Frame (JPEG marker)
SOI	Start of Image (JPEG marker)
SOS	Start of Scan (JPEG marker)
SSA	Set-Splitting Algorithm
TCAM	Ternary Content Addressable Memory
TCP	Transmission Control Protocol
TIFF	Tagged Image File Format
TTC	Toronto Transit Commission

URL	Uniform Resource Locator
VD	Variational Distance
VLC	Variable Length Codeword
WED	Weighted Euclidian Distance

Chapter 1

INTRODUCTION

Historically, prior to the invention and advent of affordable consumer computing equipment, documentation of the sexual exploitation of children was done mainly through two mediums: chemical film-based motion video or chemical film-based still frame photography, both of which were often costly to develop and difficult to distribute. As it is done today, these produced materials were then relinquished for monetary exchange or equal barter to a vast array of interested parties, thus further promoting child abuse and exploitation, and eventually creating an entire industry. This problem continues to grow rapidly, and a single instance to reflect the severity of this problem is shown from a February 2006 report by the Danish National IT and Telecom Agency. It states that of the 2,000 web addresses blacklisted for containing child pornography by Danish Internet Service Providers, an average of 36,000 daily attempts were made by Internet users to access those very same sites [1]. Table 1-1 illustrates some Internet pornography statistics for 2006.

The sexual abuse of children has long been a dark and disturbing facet of human history dating back many centuries, an issue which is large and complex, and drastically escapes the scope of this thesis. The purpose of this thesis is to primarily focus on the current problem: the electronic distribution of images which portray the illegal sexual interference of children. Current detection and prevention methodologies, legal aspects, privacy issues, and the details of an efficient technical implementation which facilitate the detection of illicit materials, transferred through data networks, are also discussed.

Internet Pornography Statistics	
Pornographic websites	4.2 million (12% of total websites)
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
Sexual solicitations of youth made in chat rooms	89%
Youths who received sexual solicitation	1 in 7 (down from 2003 stat of 1 in 3)
Worldwide visitors to pornographic websites	72 million visitors to pornography: Monthly
Internet Pornography Sales	$4.9 billion

Table 1-1: Internet pornography statistics in year 2006. (Table courtesy of http://internet-filter-review.toptenreviews.com/)

The problem this thesis approaches is the electronic transmission of child pornography. In an attempt to accomplish this, our approach uses a Stochastic Learning Weak Estimator, and a Maximum Likelihood Estimator coupled with Linear Classifiers. Our experimental results showed that the Stochastic Weak Learning Estimator coupled with a

Non-negative vector similarity coefficient-based distance algorithm was best suited towards host-based environment since its accuracy proved to be better than any other method. A Maximum Likelihood Estimator coupled with a Variational Distance algorithm showed better results for network layer approaches as its performance was superior to any of the other methods this thesis explored. For this experiment, our results showed an overhead of 16% on our router.

This thesis is organized into the following sections; Section 1 discusses the existing situation with regards to the demand, production and distribution of child images. This section also deals with law enforcement's reaction, the various agency collaborations and the issue of privacy. Section 2 reviews related works which discuss the various image recognition algorithms, and other commercially available products. The main products currently available for this process are host-based, network-based, hardware-based and combination of these three. Section 3 discusses the design and implementation of the developed software through the various stages of this process. These design stages involve image extraction, image decoding, and image classification which are explained in detail. Section 4 discusses the simulation results obtained by testing the system under varying constraints. Section 5 concludes the thesis and discusses potential direction for future work.

All images used in our experiments were legal. No actual child pornography images were ever collected, seen, sought, used or considered.

1.1 Current Situation

The current situation involving child pornography extends beyond that of a few individuals acquiring and collecting such items for personal reasons, to that of a global underground market extensively driven by supply and demand.

1.1.1 The Demand of the Market

Currently, millions of images are being electronically transmitted from sellers to buyers, for currency or barter, each day over the Internet. This trade has significantly blossomed from a niche market targeting certain demographics, to a more generalized form which strives to reside within an already accepted adult pornography ecosystem. The global reach of the Internet has certainly been the key catalyst in facilitating this market shift and has made the trafficking of this media easier than ever. It has become much easier to produce, distribute and trade. Figure 1-1 and Table 1-2 illustrates the top pornographic revenues worldwide for 2006.

Figure 1- 1: 2006 Worldwide Pornography Revenues (Graphic courtesy of http://internet-filter-review.toptenreviews.com/)

Country	Revenue (Billions)	Per Capita
South Korea*	$25.73	$526.76
Japan	$19.98	$156.75
Finland*	$0.60	$114.70
Australia	$2.00	$98.70
Brazil*	$0.10	$53.17
Czech Republic*	$0.46	$44.94
US	$13.33	$44.67
Taiwan	$1.00	$43.41
UK	$1.97	$31.84
Canada	$1.00	$30.21
China*	$27.40	$27.41
Italy	$1.40	$24.08
Netherlands	$0.20	$12.13
Philippines*	$1.00	$11.18
Germany	$0.64	$7.77
Russia*	$0.25	$1.76

Table 1-2: 2006 Worldwide Pornography Revenues in billions. (Table courtesy of http://internet-filter-review.toptenreviews.com/)

* Incomplete

1.1.2 The Production

Practically, those who produce child pornography use readily available and affordable digital imaging equipment to record images of abuse. The intense competition among the various digital equipment manufacturers in recent years has yielded lower cost devices to

enter low to medium income homes and has revolutionized the photography market. Private chemical film development labs which require expertise, specialized equipment and considerable skill are no longer required to produce professional photographs; these digital devices have greatly simplified the production and distribution of photographic images. The declining cost of complex computing equipment is made possible by Moore's Law, which factors the accelerating number of transistors in a particular unit against time. Figure 1-2 depicts a graphical representation of Moore's Law.

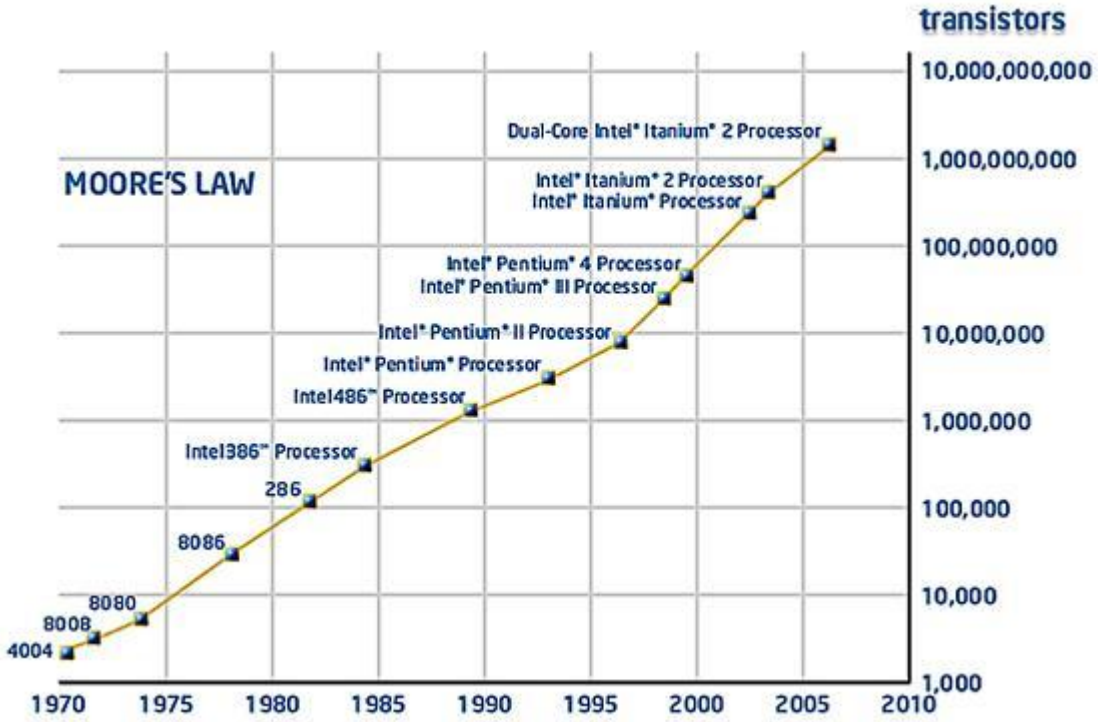


Figure 1- 2: Moore's Law (Graphic courtesy of Intel Corporation http://download.intel.com/pressroom/kits/events/moores_law_40th/)

When these images are created digitally, they require mass storage for a given amount of data. As of 2002, the average cost per gigabyte of magnetic storage was 850% less than the same in 1995 [2]. Digital photographs and digital video collections occupy tremendous amounts of space, of which the cost is reduced over time. This benefit for the

majority has led the minority of criminals having better assistance for their operations with nearly infinite storage capability, and giving further growth to their actions. Therefore, storage, which is part of the production process, has become a non-issue for the facilitation of child pornography. Figure 1-3 illustrates the cost of magnetic storage with a logarithmic plot against time.

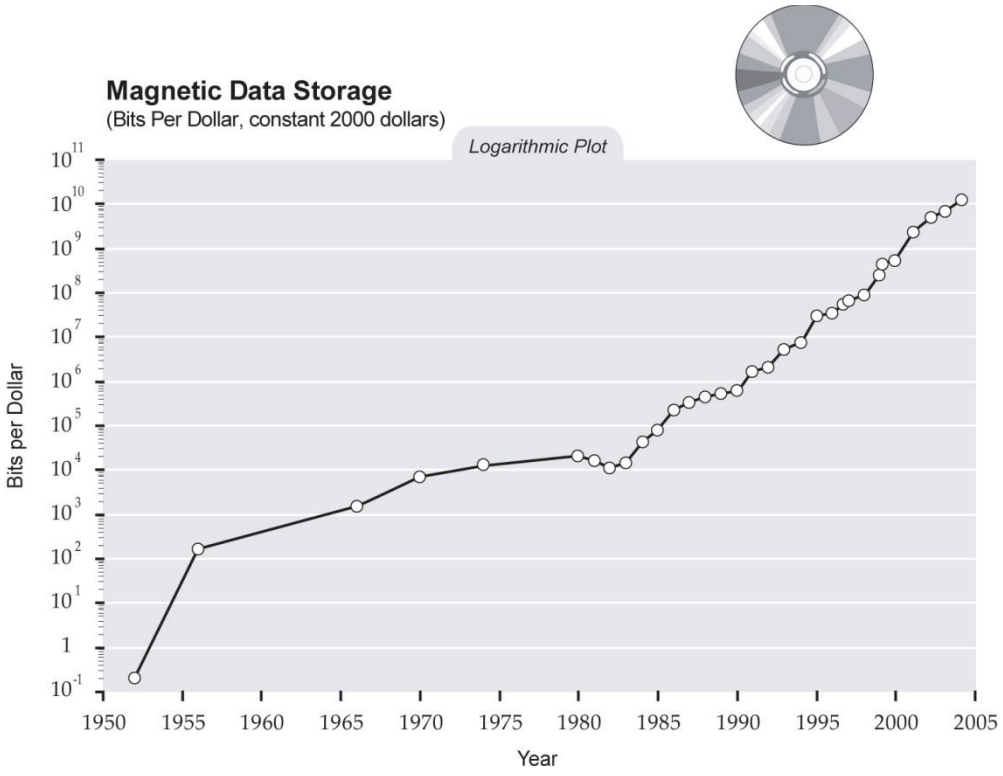


Figure 1- 3: Evolution of magnetic data storage (Graphic courtesy of http://www.singularity.com)

1.1.3 The Distribution

The distribution of these materials over the Internet has also become as simple as its production. Hundreds of peer-to-peer (P2P) services (LimeWire, Aries, Kazaa) are available to use free of charge and high speed Internet access has infiltrated nearly every industrious population worldwide [3]. These types of networks allow the free trade of

data between users, with specific content selected by the users themselves. A secondary back-end service catalogues and journals the content being shared, and then compiles this data within a central database to increase searchable targets and access. After an initial search is queried, users are directed to the results of that particular search string, these results often provide the preliminary doorway in establishing contact with those who distribute illicit sexual content. Figure 1-4 shows a screen capture of the LimeWire Peer-to-Peer application.

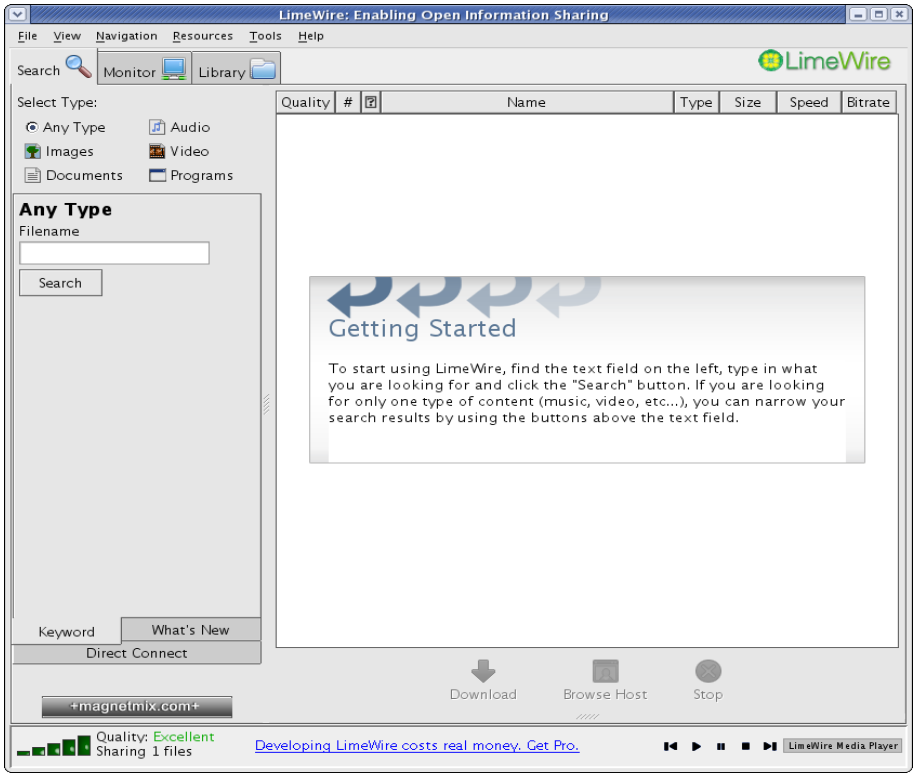


Figure 1- 4: Snapshot of LimeWire application

Internet chat rooms are another avenue of social interaction and have facilitated interactive communications for many years [4]. Users using this type of service are able to interact in real time with others who share their interests. These interactions can be done through text messages, live audio or live video. An arrangement may then be made to trade materials, illicit or otherwise, through secondary methods after contact has been

established. In March of 2006, the United States Department of Justice arrested 27 individuals engaged in the sexual molestation children using such chat rooms and interactive methods [5]. This example is one case in hundreds which have successfully been prosecuted, yet thousands still go unnoticed, mostly perhaps because of the limitations of current investigative techniques and technologies.

Traditional hypertext based websites also pose a serious challenge for law enforcement agencies. Canadian authorities estimate that over 100,000 traditional websites which contain hundreds of thousands of sexual abuse images are currently in operation [6]. These websites often sell memberships to individuals on a time-limited basis, in which the user is free to browse and download the site's content. This method of illicit material propagation remains the most simple and widely used profitable distribution mechanism, and seems to be rapidly escalating.

In total, of the millions of child pornography images seized within Canada, only 50 children have been identified by authorities as of 2003 [6]. This statistic once again emphasizes the technical boundaries and limited man power dedicated to the elimination of this disturbing and overwhelming problem.

1.2 Law Enforcement's Reaction, Operations and Activities

Law enforcement agencies are continually challenged by the burden of painstakingly ascertaining, tracking, identifying, and capturing child pornographers. Due to the number of images being produced and distributed, and the limited man power available, it is

apparent that computerized assistance must be equated within the solution. Though difficult, there have been a limited number of arrests using sting operations and collaborative efforts amongst concerned agencies.

1.2.1 Sting Operations

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 [7].

Once these identities have been established, further investigations of the individuals are conducted. 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 [8].

1.2.2 Agency Collaboration

A large amount of resources are needed to conduct operations such as *Pin* and *Candyman*, and almost all are the result of intensive collaboration between law

enforcement agencies using integrated systems and extensive communications. One such effective system developed by Microsoft Corporation and the Royal Canadian Mounted Police, allows registered law enforcement agencies to gather and analyze identification information of suspected pedophiles using sophisticated data sharing systems. The Child Exploitation Tracking System, or CETS, was officially launched in April of 2005 [6]. Previously, information detailed and gathered was inaccessible to other agencies who may have been tracking the same individuals. This issue was certain to cause a lower arrest rate and entirely ineffective against the wide outreach of the Internet. With the implementation of CETS, agencies across the world can access and update a database of information pertaining to particular persons of interest who may be involved in the production or distribution of child pornography. At present, every law enforcement agency within Canada is utilizing CETS to assist with their investigations [6].

1.3 Privacy

A vital and serious issue encompassing the detection and prevention of electronic child pornography transmissions remains the protection of privacy. As with any invasive method of analyzing data, the risk of abuse or inadvertent exposure is an element which must be addressed, and has been done so in this context in accordance with laws and regulations concerning the inspection of transient data.

To collect, store and analyze any type of data transmitted over the Internet by end-users opens a discussion and debate into the legality of such a practice, and to be certain that such methods are performed transparently and legally, a look into previous cases involving similar circumstances helps to gain a better understanding of this dilemma.

In one such case, the Toronto Transit Commission (TTC) implemented a Closed Circuit Television (CCTV) camera monitoring system into its transit system in 2004, in an effort to reduce crimes aboard civilian transport vehicles. This system operates by recording passengers as they enter or exit busses or subways, as well as for the duration of their journey, then, at one point, stored this data for up to 30 days for review if needed [9].

Privacy complaints against the TTC argued that no studies showed that such a video surveillance system would lead to a reduction in crime, and that such systems were plagued with technological faults which prevent their effectiveness. This complaint in particular was resolved when the TTC decided to erase its collected surveillance data every 72 hours, encrypt the storage mediums on which this data was stored for that period, and voluntarily agreed to commit to an annual audit [10].

To protect against unlawful personal intrusion, effective surveillance systems work to ensure that the impact from these systems are justifiable and the benefits of such system are clear, apparent and outweigh the drawbacks [10]. For example, automated teller machines, which record a user's photograph at set intervals for the safety of customers and prevention of theft and fraud [11].

Therefore, to effectively show that crime will be impacted and reduced by a specific surveillance system, a study citing justifications is undertaken to find facts and benefits, conducted wherever the system is to be implemented. Of note is that such guidelines of justification, conformity and methods of collection or retention do not apply to covert

surveillance operations involving court-warranted investigations by law enforcement agencies. Yet, for all other systems, in sum, it must be shown that a reduction *and* prevention of crime, thereby also rendering it as an effective deterrent against criminal activity [10].

Laws which apply specifically to user data have been enacted, amended and legislated within the Canadian *Privacy Act* and within the *Personal Information Protection and Electronic Documents Act* (PIPEDA). Amongst its complexities, these legislations make it specifically clear that law enforcement agencies have the right to query carriers for data when certain criteria have been realized, amongst these are, Section 7, 3.2 of the PIPEDA legislation states [12]:

7. (3) (c.1) Made to a government institution or part of a government institution that has made a request for the information, identified its lawful authority to obtain the information and indicated that

(i) It suspects that the information relates to national security, the defense of Canada or the conduct of international affairs,

(ii) The disclosure is requested for the purpose of enforcing any law of Canada, a province or a foreign jurisdiction, carrying out an investigation relating to the enforcement of any such law or gathering intelligence for the purpose of enforcing any such law, or

(iii) The disclosure is requested for the purpose of administering any law of Canada or a province;

This section of the PIPEDA legislation makes it clear that law enforcement authorities who have obtained valid warrants are privy to communications and materials obtained through communications channels by request, and in this case carriers who facilitate the communications.

Furthermore, current Canadian laws enable law enforcement to make requests to telecommunications carriers who are, by law, obliged to facilitate communications intercepts. In the United States of America, the *Communications Assistance for Law Enforcement Act* (CALEA), specifically details the exact obligations of carriers to build monitoring capabilities into their service architectures, for the purposes of lawful interception. This architecture is present in all forms of commercial communications architectures as it is required by law [12].

Unfortunately, this type of facilitation is not yet available in Canada by way of legislation. Since monitoring capabilities are present within these systems, the Provincial and Federal Privacy Commissioners have set out a framework to minimize abuse of these capabilities. Privacy by design is a concept which underlines the most basic principles of data protection and broadly extends them to apply to technologies capable of surveillance, such as CCTV and telecommunications, voice, data or otherwise [10].

In the context of detecting and preventing electronic child pornography transmission, using the system detailed in this thesis, no personally identifiable data is collected or stored indefinitely, nor is it forwarded on to any systems which perform storage. Rather,

this system works by actively monitoring data packets for child pornography by processing data in real time and collecting statistics, once certain preset thresholds are achieved, the operator of the equipment may notify law enforcement of such activity, who may then seek valid warrants to perform further monitoring. They can also begin data collection, or identify the source and destination of the transmissions, and then take the necessary actions against the individuals or groups responsible for the illicit traffic.

Our network-based approach occurs within two distinct stages to determine how exactly monitoring and alerting occurs. In Stage I, while the system actively monitors data, an obscenity score is calculated according to each connection, but, no personally identifying data is recorded. At Stage II of our system, if the threshold for the obscenity score is exceeded, the system will begin to record and store additional information about that particular connection, including IP address and packet size information.

Privacy in mind, Association Xpertise Inc. has created a code that contains 10 principles of fair information practices, which form ground rules for the collection, use and disclosure of personal information [34].

The 10 principles are:

1. ***Accountability:*** Who is in charge of audit, evaluation and investigation?
2. ***Identifying purposes:*** Why is this information collected?
3. ***Consent:*** Upon whose authorization is this information being collected?
4. ***Limiting collection:*** What is the minimum required amount of information collected?

5. ***Limiting use, disclosure, and retention:*** How will collected information be used, to whom will it be disclosed, and how long must it be retained?

6. ***Accuracy:*** How accurate is this system?

7. ***Safeguards:*** Has the collected information been secured?

8. ***Openness:*** What is our personal information policy and practices?

9. ***Individual access:*** What are the policies to access personal information?

10. ***Challenging complaints:*** How do we investigate complaints?

For our work we use these 10 principles as a basis of policy during the creation of a prototype that is capable of intercepting and classifying traffic as it passes though a router. Table 1-3 shows the 10 PIPEDA principles and how we met these requirements.

PIPEDA Requirements	**Prototype System**	
	Stage I (while monitoring)	**Stage II (upon triggered alarm)**
Accountability	Limited accountability since no personal information is retained.	The existing Privacy Officer for Law enforcement organizations will be accountable for any issues.
Identifying Purposes	To check for any illicit images being transferred across the network.	To check for any illicit images being transferred across the network. Record the source and destination addresses of suspected traffic for further investigation.
Consent	The consent of the user is not required since no information is recorded.	The consent of the user is not required since law enforcement may have obtained a warrant at this stage.
Limiting Collection	No information is recorded; the system stores the obscenity score.	Only suspicious IP addresses and packet size information is stored.
Use, disclosure,	No information is recorded.	Information is collected only for suspicious IP addresses and only law

retention		enforcement agencies have access to it.
Accuracy	The accuracy of our system will be determined after the implementation of the system in a real network setup.	
Safeguards	There would not be any information to safeguard.	Any information collected or stored is kept with law enforcement agencies using existing safeguards.
Openness	Under no circumstances does our system collect any personal information.	Source and destination IP of suspected traffic will be recorded.
Individual Access	Individual requests must conform to current Access to Information guidelines.	Individual requests must conform to current Access to Information guidelines.
Complaints	Complaints against this system will be handled by the ISP.	Compliance factors are judged by the corresponding law enforcement agencies as well as a government privacy watchdog group.

Table 1-3: PIPEDA principles and personal response

1.4 Prototype Design

The system is composed of two main parts – a host-based component and network-based component. In the network-based component, we have a packet level system which classifies images passing through the network using Linear Classifiers as is detailed in section 3.5. The network-based component has the advantage of working with both compressed and uncompressed images. For the host-based component, we have an image matching system that is capable of searching for images that are already black-listed. The feature vectors of the black-listed images are calculated using the RGB colour space (see sec.3.6). Our system has been tested using the MLE and SLWE algorithms independently and these have not been tested to work in unison.

Figure 1-5 depicts a general architecture of the two approaches in relation to our image matching and classification system.

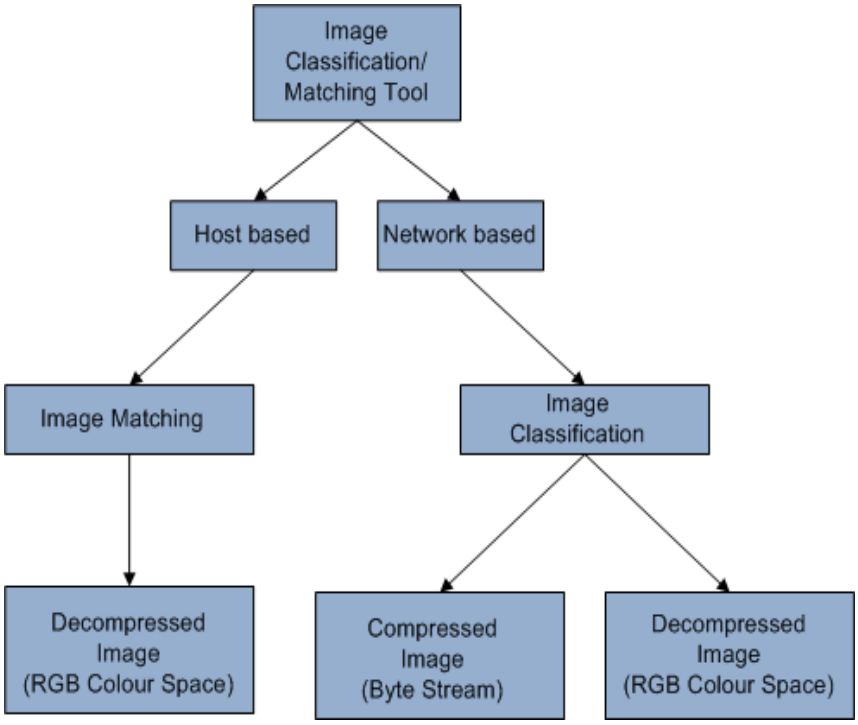


Figure 1- 5: Flow diagram of our system

Chapter 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. Of these works, none offers a total solution as of yet, however, cannot be discounted as they are certainly a crucial element to the overall solution.

2.1 Image Recognition

Image Recognition software has been under intensive development for over three decades and is used in many industries to assist users with the tedious task of identifying images when a specific criterion is presented.

A widely used method of image recognition termed as Content-Based Image Retrieval (CBIR), operates by analyzing the colour and geometry of images. This approach uses fundamental visual properties including background, texture, and shape. The information is then used to produce queried results based on the skin content of a picture that are above a certain threshold [13].

CBIR greatly reduces the time needed to process images. As an example of this, the Discrete Cosine Transform (DCT) algorithm uses features of image intensity, attained to extract ordinal descriptors of the image to construct an index for faster responses [14].

Although much research has been done in the area of CBIR, little progress has been made to fully implement an engine solely based on the search of image content. Most CBIR systems are deficient in fully incorporating low level features of images, such as intensity, colour, texture, shape and spatial constraints characteristics, with the high level features such as semantic content.

Hove et al. [31] proposed a CBIR system to deal with this issue that uses a shape thesaurus, instead of text thesaurus, to link the low level features extracted with semantics. Compared with text thesaurus, shape thesaurus can inject more features into CBIR related techniques. Thus, the system offers better efficiency by combining shape information with the text thesaurus.

Feng et al. [32] also proposed the chain code as a principle method to represent shape descriptors. In this way, new definitions can be deducted from existing ones. The reason that the chain code is used to create the shape thesaurus is that it is simple to use the compared descriptors with other methods, such as Fourier Shape Descriptors (FSD) and Moment Invariants (MI). We use FSD, MI, Finite Element method (FEM), Turning Function and Wavelet Descriptor to obtain image shape to compare with other images' shape thesaurus.

Che-Jen et al. [29] 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 Direct Current (DC) coefficients to search for suspicious files. Different DCT coefficients adopt a

variable length of codes as baseline JPEG files use Variable Length Codeword (VLC) encoding. Each Minimum Coded Unit is accountable for extraction of Y, Cb, Cr in a sequential manner. End of Block (EOB) values allow extraction of the DC components of the next block. Extracted DC components from JPEG packets with the same feature database are used for comparisons. The problem with this approach is that the system 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. [30] 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) colour space.

Forsyth et al. [42] 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). David used a statistical analysis of data settings to further refine the ranges, as optimal parameters. Skin tone defined as any pixel with Hue values between 2 to 18 (of a maximum 180) and saturation values between 80 and 230 (of a maximum 255). Statistical analysis of data settings can be used again to refine the ranges.

The system divides the number of skin tone pixels within skin tone range in the Internet object image by the total number of pixels in the image and compares the resulting value against an automatically generated or user-defined threshold value. Those images

resulting with a ratio less than the threshold proportion of skin pixels are discarded. Images that are equivalent or greater than the threshold proportion of skin pixels are then assigned a nudity coefficient which is equal to the percentage of skin pixels in the image. Factors like the size of the image and encoding type are used to weigh the nudity coefficient of each image in an object.

Similarly, the WebGaurd [41] 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 a feature vector. To improve performance, an analysis using a skin colour pixel mode is used.

2.2 Currently Available Products

A large selection of products designed by companies to reduce the flow of illicit material into homes and businesses are available and affordable to the average consumer. They may be application or hardware-based and can vary greatly in their function and effectiveness. In particular, 95% of schools, 43% of public libraries and 33% of parents in the United States employ some variant of filtering technology to block content deemed inappropriate such as pornography [15].

Of these products, they are normally classified into two main categories, host-based, or proxy-based. These two categories are further divided by their specific method of operation into two distinct forms, static and dynamic filtering, which maybe used individually by a product, or can be used in conjunction.

2.3 Dynamic and Static Filtering

Dynamic filtering is a proactive method of restricting content based on predetermined variables such as written content and website address. This type of filtering uses a more complex method of restricting access to inappropriate materials than static filtering provides. It uses a complex dictionary-based analysis of text to determine if the content is suitable or unsuitable for viewing.

Static filtering uses a predetermined blacklist method by blocking content using specific keywords within web pages or their addresses, of which can also be blocked separately (example – Cleanfeed, NetNanny®).

2.4 Host-based and Proxy-based Filters

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® [16] and Cyber Patrol® [17].

Proxy-based filtering is 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 connect through this server which filters websites previously blacklisted by addresses or keywords and

restricts access to these sites and their content. Generally, because of load considerations, this particular type of filtering uses a static method.

2.5 Network Approaches

Existing products and concepts concerning packet classification have been researched and developed, and analysis of their effectiveness is currently available. Most of these systems perform operations by analyzing packet header data, and are focused towards intrusion detection and prevention systems, or quality of service (QoS) features.

For QoS applications, routers have taken the task of admission control, weighted queuing, and resource scheduling. These mechanisms work by distinguishing packets, with data types extracted from headers, and making processing decisions based on preset rules. Gupta [18] summarizes certain solutions using packet classification based on a criteria applied to the packet header using basic search algorithms (for example: Linear Search, Caching, Hierarchical Tries, Set-Pruning Tries), and geometric algorithms (for example: Grid-of-Tries, Area-based Quad-trees, Fat Inverted Segment Tree), and heuristic algorithms (for example: Recursive Flow Classification, Hierarchical Cuttings, Tuple-Space search), and hardware-specific search algorithms (for example: Ternary Content Addressable Memories, Bitmap-Intersection), with each of these algorithms applied on the IP header of a packet.

Feldman et al. [19] proposed the Fat Inverted Segment (FIS) tree based algorithm for implementing multi-field range based classification, with a limitation to the nature of FIS

trees having to conform to classification searches multiple times in order to successfully traverse a FIS tree.

Wang et al. [20] also mentions another approach intended for QoS consisting of a disjoint based algorithm for a multi-field range based packet classification. This algorithm uses a new data structure of disjointed graphing sets composed of multiple Elementary Interval Trees, and Disjoint Interval Trees, representing a given set of rules, where only a single path of traversal is required during packet classification searches.

Warkhede et al. [21] has taken a different approach to the previously mentioned ones, and presents a multi-dimensional encoding factor, focused upon spacing comprised of the source and destination IP address, source and destination port and protocol type. This type of encoding style by dividing rules, set into several multi-dimensional collision-free rule sets, which are adopted to create new coding vectors known as Layer Coding Vectors. Therefore, in regards to efficiency and resource usage, tests conclude that these methods increase efficiency and reduce resource requirements.

Network Intrusion Detection Systems (NIDS) require packet-level multi-match classification, whereby each packet must be checked against all rules. Ternary Content Addressable Memory (TCAM) has been adopted to solve multi-match classification issues because of their abilities to perform fast parallel matching. Yet, TCAM based approaches are notable, since the very nature of CAMs require a full search of stored words spanning the entire available addressable range, as opposed to Random Accessible

Memory (RAM), which obtains memory addressing information from the operating system itself. TCAMs are adopted to solve multi-match classification issues, but are subject to large power consumption and resource costs. Because of fast parallel matching advantages, TCAMs are primarily used within solid-state computing devices, such as routers [22].

For NIDS systems, software-based schemes of performing classification are noted. To resolve general CAM inefficiencies, Yu et al. [23] proposes a novel scheme that overcomes these CAM related issues by using a new Set-Splitting Algorithm (SSA). The design of SSA allows it to split filters into multiple groups and performs separate TCAM lookups into these groups. Packet-level approaches such as NIDS require the usage of multi-matched classification, whereby matching filters are reported. It works to guarantee the removal of at least half the intersections when a filter set is split into two different sets, resulting in lower TCAM memory and resource usage [23].

Thus far, the previously described classification methods of QoS and Network Intrusion Detection Systems use IP packet headers to classify packets, and few attempt to determine content based on actual payload, with the exception of certain Deep Packet Inspection systems and limited Network Intrusion Detection Systems.

Wang et al. [24] proposes such a Network Intrusion Detection System which uses the payload of the packet itself, and processes it within an anomaly detector. Its efficiency is of note as well as the automated and unsupervised manner in which it operates. Profiling

of byte frequency distribution and the standard deviation of the application payload, flowing to a single host and port, is provided during the training stages. When performing classification, the Mahalanobis Distance calculation is used for the detection of similarities between new data and the pre-computed profile, then compares this calculation against a measured threshold and generates an action when the distance of the new input exceeds that of the defined threshold.

Each of these methods works to meet the definitions and expectations for their intended usage and many work well enough to accomplish their specific tasks. The specified detection of image or video content in a granular method within a packet, and its rapid analysis and profile computation, are difficult to accomplish in a network-based approach. None of these approaches provide an integral solution; however they all represent genuine efforts towards the eradication of obscene material.

Of these, one proposal uses 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 [25-27]. During the training stages, pre-labeled IP packets are input sequentially from symbols drawn from an alphabet, and taught to learn these statistics using a Stochastic Weak Estimator. This also takes into account the variability of the distribution source. The classification rule, once it is derived, is then tested to analyze its performance, and this validation is undertaken by enabling a testing set with pre-labeled IP packets into their linear classification rules, and then recording the accuracy of the classification data. This is done *n* times with an *x* amount of testing sets

derived from partitioning the entire data set, in a process of cross validation. When classifying packets from the data set, the Stochastic Weak Estimator is also used to extract the relevant features of the inputted classification rules. After this, the linear classification rule labels each byte of the packet with preset criteria, such as pornography or otherwise, and the entire packet is deemed as pornography if the majority of its bytes are labeled as such.

2.6 Hardware Devices

Hardware-based appliances dedicated to securing perimeter networks typically make use of URL and keyword filtering at the TCP/IP Layer 3. These types of devices, though usually more expensive than software solutions, are designed to provide efficient and uninterrupted traffic processing under maximum loads, and can accommodate the most complex keyword and URL filtering requirements [28].

Though these solutions are effective for 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. And finally, hardware-based filtering is easily overcome by gaining knowledge of websites which do not use expected wording or URLs for their sites and content.

Chapter 3

DESIGN AND PROTOTYPE IMPLEMENTATION

The system is composed of three major blocks – image packet interception, image decoding and image classification. For packet interception, we used Netfilter (see sec. 3.1), a subsystem in Linux 2.4. Our system uses Netfilter for packet classification because of its open-source framework, its efficiency. It also provides a high level of programmability which features well defined functions that allow advanced parameters for packet interception.

Shown in Figure 3-1 and 3-2 are the diagrammatic representations of the overall system – the first is for a compressed data stream (byte stream) and the latter for decoded RGB colour space images (see sec. 3.2 for JPEG image coding and decoding). For our system, we adapted a JPEG-based decoder due to a very large portion of Internet images being comprised of the JPEG file format. The input to the system is the Internet traffic which consists of packets. As each packet is captured by the system the image filters are engaged and processes the packets that contain images (see sec. 3.3). The image packets are then passed through the binary JPEG image assembler which constructively adds the data in each packet to form the binary sequence for an image. This binary sequence is then sent into the JPEG decoder which converts the binary sequence to its RGB components. The feature extractor extracts information based on two algorithms (see sec. 3.4), the Maximum Likelihood Estimator (MLE) and the Stochastic Learning Weak Estimator (SLWE). The obtained features are then compared against two feature vectors, which are obtained during the training stage of the system (see sec. 3.5).

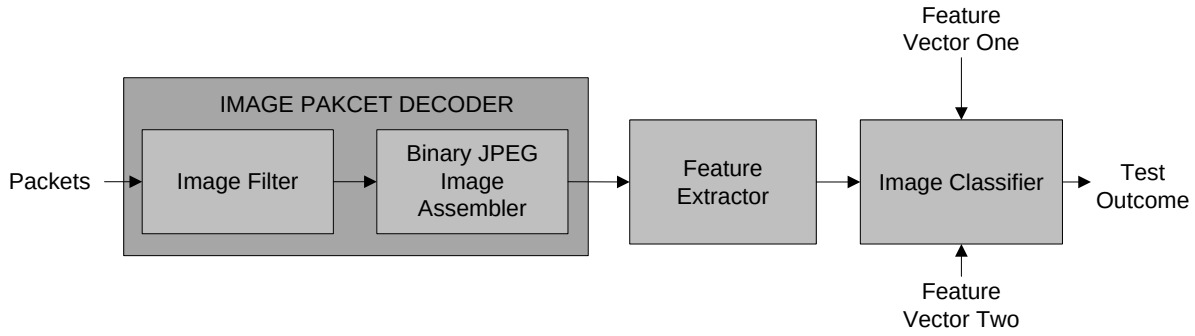


Figure 3- 1: Block diagram of proposed method for byte stream data.

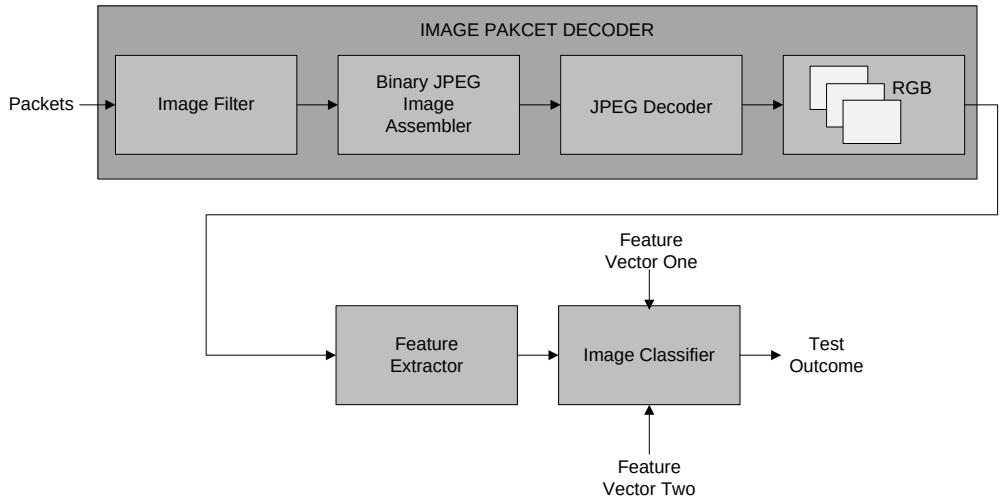


Figure 3- 2: Block diagram of proposed method for RGB colour space data.

Figure 3-3 illustrates the process of classification where feature vector *one* and feature vector *two* are obtained from the training stage, and for any image which passes thorough our system, we calculate its feature vector. The distance, $d1$, is a distance between the test image's feature vector and nude images, and $d2$ is the distance between the test image feature vector and feature vector of non-nude images. The end result is the classification of the image being illicit or non-illicit. Each module of the system is explained in detail below.

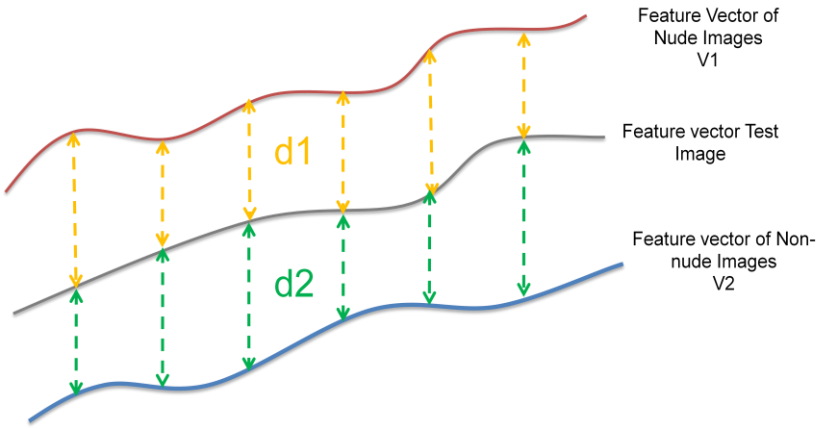


Figure 3- 3: Classification process of proposed method

To accomplish image classification at the network layer, traffic capture was required. Network analysis was done using Netfilter to capture each packet transiting the network. Section 3.1 below provides a detailed explanation.

3.1 Netfilter

The Netfilter module is used to intercept and capture packets. Netfilter is a subsystem in the Linux 2.4 kernel which facilitates packet filtering, network address translation (NAT) and connection tracking, possible through the use of hooks in the kernel network code. Hooks are defined as places in the kernel code (either statistically built or in the form of a loadable module) that can register functions to be called for specific network events. As a packet moves through the Linux Kernel network stack, it passes through several hook locations where these packets can be analyzed and kept or discarded. More information on Netfilter can be found in [39].

The Linux kernel starts with three lists of rules in the filter table which are called chains or firewall chains**.** These three chains are INPUT, OUTPUT and FORWARD. A chain is

a checklist of rules and if a rule does not match the packet, then the next rule in the chain is referenced and executed. The kernel looks at the chain guidelines to make a decision if there are no more rules to consult. In general the guideline tells the kernel to DROP the packet. Figure 3-4 illustrates the Netfilter chain operation and Figure 3-5 details the hook function of the Netfilter architecture.

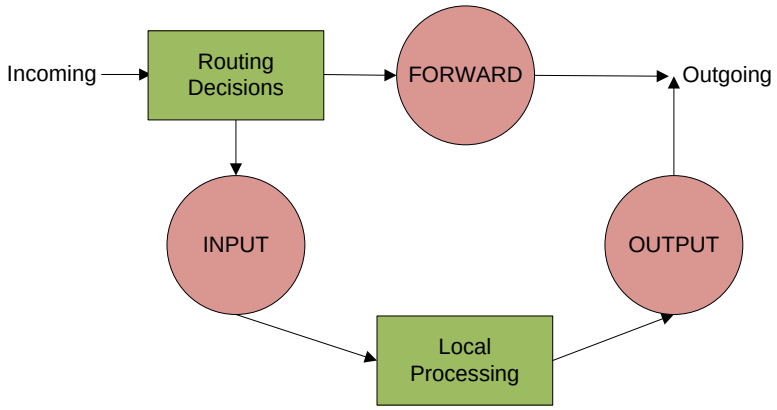


Figure 3- 4: Netfilter chains: INPUT, OUTPUT and FORWARD. (Slightly modified version: original version from Netfilter.org Project [39])

An explanation of Netfilter mechanisms:

1. Routing: The kernel first looks at the destination of the packet on its arrival.

2. If the destination matches the box it is routed downward to the INPUT chain.

3. If forwarding is disabled or if the kernel does not know how to forward the packet it will drop it.

4. A packet is routed to a FORWARD chain if it is destined for another network interface.

5. A program running on the box can also send network packets through the OUTPUT chain if it is accepted.

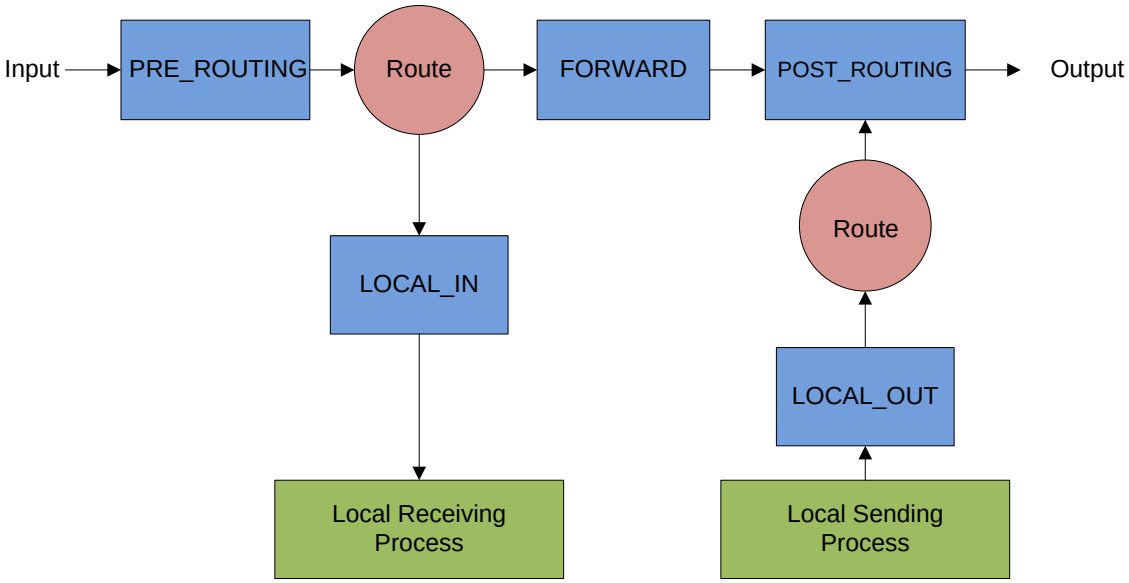


Figure 3- 5: Netfilter architecture showing points where hook can be inserted. (Slightly modified version: original version from Netfilter.org Project [39])

Netfilter defines five hooks for the IPv4 protocol. Listed below, in Table 3-1, the five possible hooks and their functions are described further.

Hook	Meaning
NF_IP_PRE_ROUTING	Observe all valid incoming packets
NF_IP_LOCAL_IN	Observe packets destined for this IP address.
NF_IP_LOCAL_OUT	Observe locally generated packets.
NF_IP_FORWARD	Observe packets being forwarded through the machine.
NF_IP_POST_ROUTING	Observe IP packets immediately before be transmitted.

Table 3-1: Netfilter hooks and their meaning

Our system makes use of the NF_IP_PRE_ROUTING hook, which is the first hook once the packet is received. Once the hook function has processed the packet, it must be

returned to with one of the predefined Netfilter codes. These codes and their definitions are listed in table 3-2.

TARGET	MEANING
NF_ACCEPT	Accept this packet. Terminating Valid in the filter table
NF_DROP	Ignore this packet. The sender gets no notification. Terminating Valid in the filter table
NF_STOLEN	Netfilter to "forget" about the packet. What this tells Netfilter is that the hook function will take processing of this packet from here and that Netfilter should drop all processing of it. This does not mean, however, that resources for the packet are released. The packet and its respective *sk_buff* structure are still valid, it's just that the hook function has taken ownership of the packet away from Netfilter
NF_QUEUE	Queue packet for user space. Packets can be further processed at the user space and sent back into network stack.
NF_REPEAT	Call this hook function again

Table 3-2: Netfilter targets and their use

3.2 Image Coding, Decoding

There are many different image format types used on the Internet such as JPEG, BMP, and TIFF images. The most popular and widely used image type is JPEG. For our implementation we only consider JPEG images and this section deals with the encoding and decoding of that format. As the system deals with the conversion of the binary sequence to a JPEG image, this section deals more with the process of decoding a binary

bit stream rather than the encoding of it. However, one must note that the decoding process is just the reverse process of the encoding process. The complete guideline on image coding and decoding can be found in [37-38, 43-45].

3.2.1 JPEG Encoding

The JPEG encoding process is performed on blocks of an image that are 8 pixel wide by 8 pixels high. Each of these JPEG data blocks is encoded in a sequence of three operations. First the image block is transformed using a 2-dimensional Forward Discrete Cosine Transform (FDCT) to determine the spectral components of the image. After the FDCT is performed the upper left corner of the coefficient matrix contains the DC component of the block and the lower right corner contains the highest frequency components of the image. Since the human eye does not readily perceive high frequency changes the high frequency components can be stored with less precision than the more important low frequency components. This low pass filtering of the image is performed by the next stage, which quantizes the data in exactly this manner. The Quantization table used in this step determines the exact filter characteristics and thus the compression ratio and quality of the encoded JPEG image. Finally the Coding stage transforms the 8×8 quantized block into a linear stream of values and then assigns the more frequently occurring values to shorter binary codes and less frequently occurring values to longer binary codes to minimize the length of the encoded message. The Coding table used in this step determines the compression ratio since the table must accurately match the relative frequencies of the input values to achieve good compression. FDCT and the Inverse Discrete Cosine Transform IDCT are defined as:

FDCT: $$S_{u,v}(u,v) = \frac{1}{4} C_u C_v \sum_{x=0}^{7} \sum_{y=0}^{7} s_{x,y} \cos\left(\frac{(2x+1)\cdot u\pi}{16}\right) \cos\left(\frac{(2y+1)\cdot v\pi}{16}\right)$$

IDCT: $$s_{x,y}(x,y) = \frac{1}{4} \sum_{u=0}^{7} \sum_{v=0}^{7} C_u C_v S_{u,v} \cos\left(\frac{(2x+1)\cdot u\pi}{16}\right) \cos\left(\frac{(2y+1)\cdot v\pi}{16}\right)$$

where $C_n = \begin{cases} 1/2 & n = 0 \\ 1 & n \neq 0 \end{cases}$, and $S_{u,v}(u,v)$ is the FDCT value of the block matrix entry (u,v), $s_{x,y}(x,y)$ is the IDCT of the block matrix row x and column y. These equations can also be further rewritten in the form of linear transformations using matrices as follows:

$$S = D \cdot s \cdot D^T$$

$$s = D^T \cdot S \cdot D$$

Where D is a constant 8×8 matrix formed by the cosine values and constants above. The JPEG decoder used this linear transformation equation to compute the IDCT since computing the product of two matrices is relatively straight forward and less time consuming.

3.2.2 JPEG Decoding

Part of our experiment was to try to classify images using RGB colour space and compressed binary images. Every image packet was extracted and decoded to get its RGB components. Thus, this section is our main concern, since any extracted image from our network has to be reconstructed, partially, or in full, so that it can be processed by the classifier. The JPEG data in packet form is entered into a network buffer which is gradually filled until the entire image is received. This data is then forwarded onto the decoder.

The JPEG decoding process is an inverse transformation where the encoded data is first decoded and restored to 8×8 pixel data blocks using a preliminary Decoding stage. The main blocks of JPEG decoding include Run-Length Decoding (VLD), Zig-Zag scan and Huffman Decoding (ZZ/HD), quantization (DQ) and IDCT. In the quantization stage, quantization table specification is used to approximately regain the spectral components of the image block, while low frequency components may be fully restored the high frequency components may be severely distorted however this distortion is barely perceptible. Finally the Inverse Discrete Cosine Transform approximately recovers the original 8×8 data block. Figure 3-6 is a detailed block diagram showing this process.

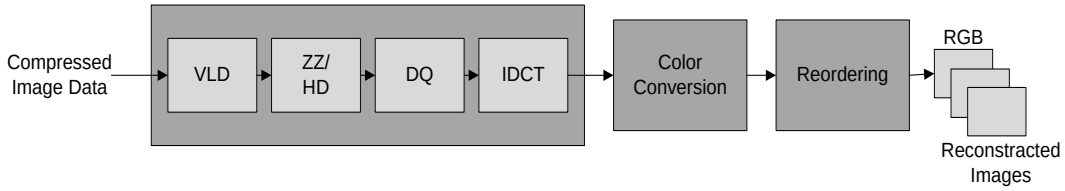


Figure 3- 6: Block diagram of JPEG decoding (Slightly modified version: original version from Stuijk [45])

The compressed JPEG image data forms a byte stream input for the decoder. This byte stream contains two-byte combination markers which identifies a structural part of the compressed image data. The first byte is always 0x'FF'. The second byte is defined in the JPEG standard (see Appendix A). This byte indicates which of the structural parts of the compressed image data follows the marker in the byte stream.

An image can be separated in a number of colour components. This results in a set of grayscale images describing the tone of the colours in the image. When an image is, for instance, separated into its red, green and blue components, you obtain three grayscale

images describing the red, blue and green tones in the image. Every grayscale image describing a tone can be divided into smaller parts using a grid of 8×8 pixels.

The JPEG standard describes the syntax for the flow of the compressed image data. The syntax for flow of this compressed image data in a baseline JPEG decoder is given in Figure 3-7.

A valid JPEG compressed image data stream always starts with a start of image [SOI] marker. After the SOI marker a number of different markers may be found. These identify for instance, quantization or Huffman tables that are needed for the decoding. After getting zero or more of these tables, a start of frame marker [SOF] may be found. After the SOF marker, these tables can also be defined. After zero or more of these tables, a start of scan [SOS] marker must be found. After the SOS, we find a number of entropy coded segments [ECS] in the compressed image data stream. These ECS contain the coded values for all pixels that comprise the image. These pixel values are grouped in so called minimal coded units [MCU].

Depending on the horizontal and vertical sampling factors of every colour component, it may be necessary to take one or more blocks of that component into a MCU. The maximum number of blocks in a MCU is however limited by the standard to at most ten. If the entropy-coded segment does not contain the last MCU of the image, then a restart marker [RST] is found after the entropy-coded segment in the compressed image data. After this restart marker, another entropy-coded segment starts. This process is repeated

until all entropy-coded segments are processed by the decoder. Then another scan may be found in the compressed image data, or the compressed image data ends with an end of image marker [EOI]. The order in which the decoder finds the MCUs is shown in Figure 3-8.

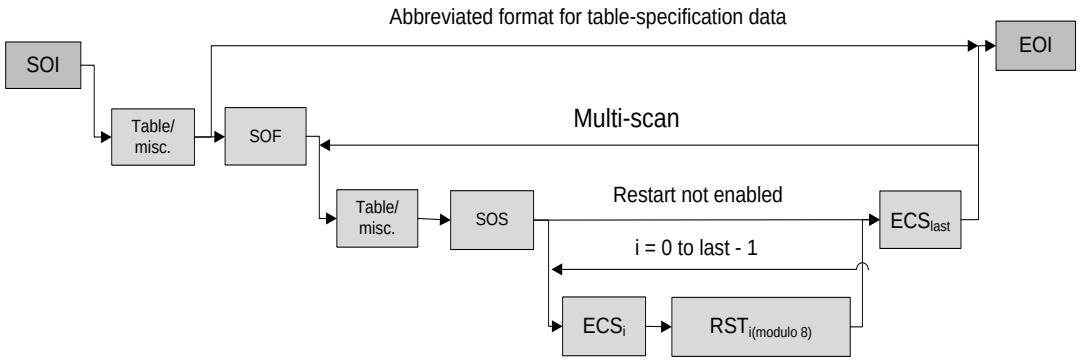


Figure 3- 7: Block diagram of JPEG bit stream decoding (Graphic courtesy of Stuijk [45])

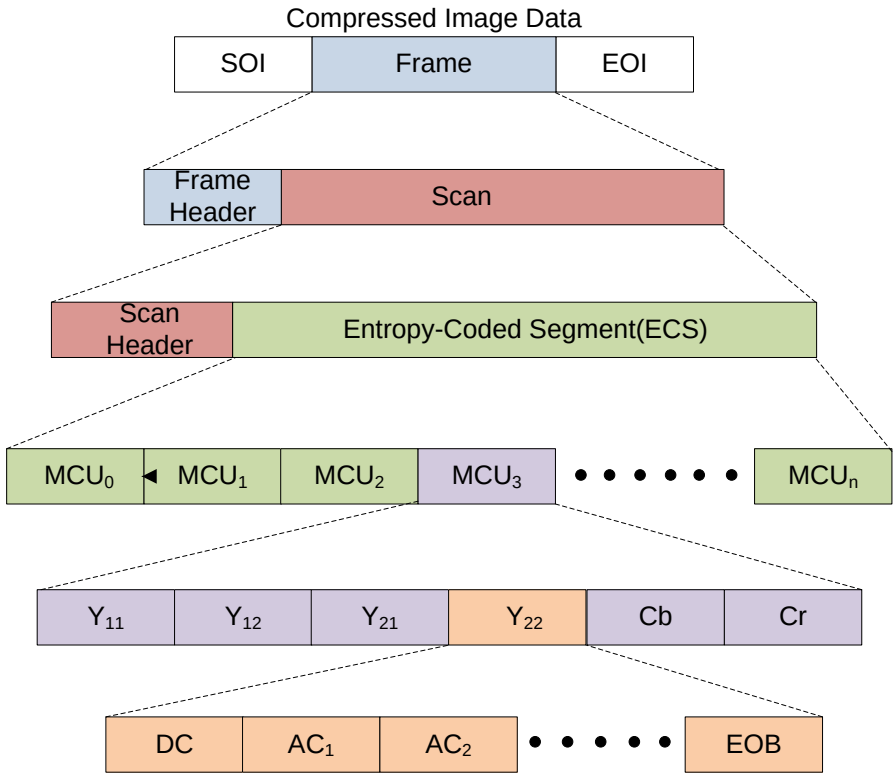


Figure 3- 8: JPEG bit stream structure (Modified version: original version from of Che-Jen [29])

3.3 Image extraction

Our assumption is that every packet which passes through our system does so through a gateway on which our system is implemented as shown in Figure 3-9. Our system then processes traffic by differentiating various classes of packets, such as image, or non-image packets. To accomplish this, 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. For our system, three buffers are sufficient to accommodate network traffic. To extract only image packets from overall network data, SOI markers are used to identify image packets, which are then placed into an available buffer. This is accomplished by filling a buffer once an SOI is detected, then emptying the buffer once the EOI is detected, and sending data onwards for further processing.

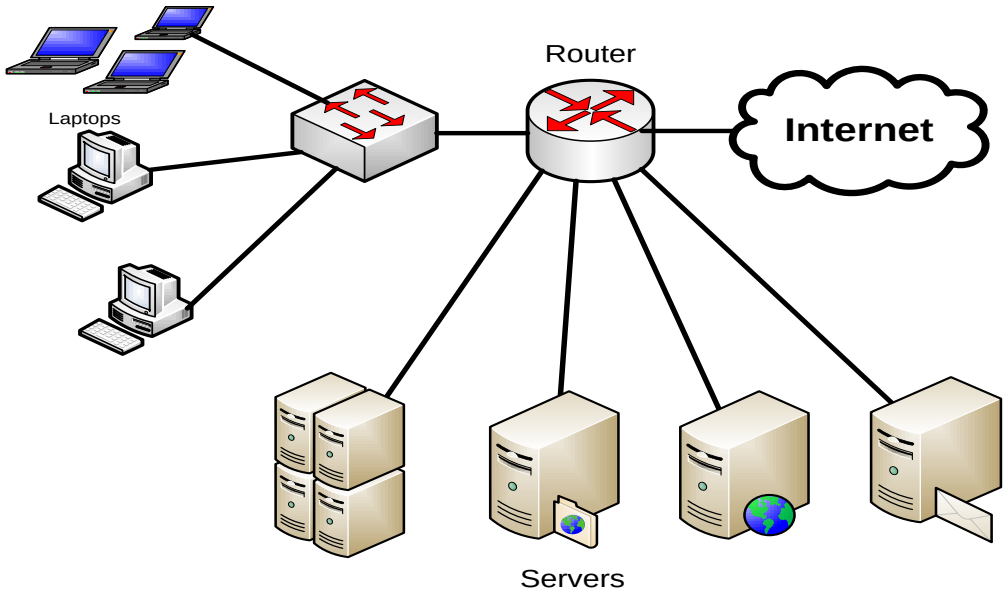


Figure 3- 9: Possible realization of our network; our system is implemented on the router. Every packet passes through the router.

TCP sequence numbers are used to extract the remaining image data until an EOI marker is found within our payload. When the buffer is full, the payload is copied to the JPEG decoder and the TCP headers are processed for further analysis if needed. Algorithm 4.1 shown below details the process of extracting image packets. In a byte stream method, the entire contents of the buffer data are directly passed onto the classification module. In RGB colour space transfers, buffer data is sent onto a JPEG decoder for decoding, and once decoded the output is the RGB colour space.

Algorithm 3.1: *imageExtract(packet)*

1. *[1] **set** *buf* = ***NULL***
2. *[2] ***set*** *buf_start_filling* = ***False***
3. set *payload* = ***extractPayload****(packet)*
4. ***set*** *size* =***size****(payload)*
5. **If** payload[0] = SOI

   4.1 **set** *buf = payload*

   4.2 **set** *buf_start_filling* = ***True***
6. **If** *buf_start_filling = True* and packet_in_sequence

   6.1 **append** (*buf, payload*)
7. **If** *payload[size-1] = EOI*

   7.1 **append** (*buf, payload*)

   7.2 **extract_image**(*buf*)

   7.3 **set** *buf_start_feeling* = ***False***

*[1] and *[2] are declared globally and initialized to *NULL* and *False* respectively.

3.4 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. Figure 3-10 shows the process of feature extraction of the RGB colour space data in the training stage. Note that MLE and SLWE are used separately.

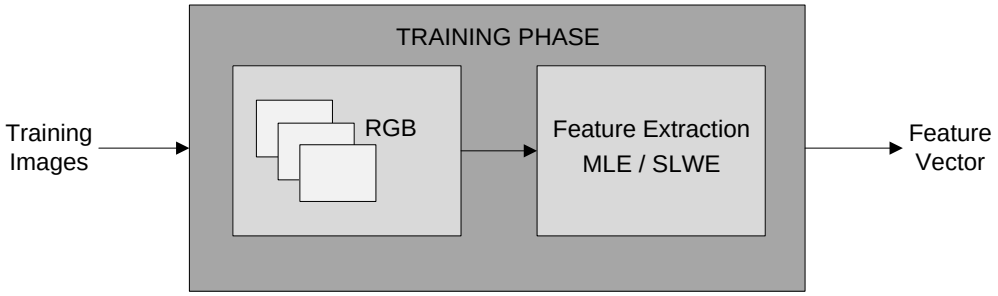


Figure 3- 10: Image feature extraction using MLE/SLWE estimators for RGB colour space

3.4.1 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 parameter estimation is to determine the parameters that maximize the probability (likelihood) of the sample data.

MLE offers a way of tuning the free parameters of the real world model to provide a good fit. 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 work

we used 256 symbols for every components of our colour space and also 256 symbols for byte stream.

The specific figure of 256 symbols was chosen because for any byte stream or RGB colour space data, the range is defined from 0 to 255. The RGB colour components are defined from 0 to 255 and the possible values for byte stream data is also 0 to 255, since we have 8-bits in one byte.

For the byte stream we want to estimate $S = [s_0, s_1, \dots, s_{255}]$ and s_i calculated as:

$$s_i = frequency\ of\ i\ /\ total\ number\ of\ bytes,$$

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

For our colour space samples, we have 256 colour realizations (symbols) of every component. Thus we want to estimate the outcome of each symbol for each colour component, namely red, green and blue. Therefore $S = \{S_R, S_G, S_B\}$ and $S_R = [s_{R0}, s_{R1}, \dots, s_{R255}]$, $S_G = [s_{G0}, s_{G1}, \dots, s_{G255}]$, and $S_B = [s_{B0}, s_{B1}, \dots, s_{B255}]$

$$s_{Ci} = \frac{frequency\ of symbol\ i\ on\ one\ of\ the\ color\ component}{total\ number\ of\ pixels}$$

and s_{Ci} is the feature vector entry of symbol i in one of the colour components, $C = \{R, G, B\}$. Algorithm 4.2 explains the MLE algorithm.

Algorithm 3.2: $\boldsymbol{MLE(Image, row, col)}$

1. $\boldsymbol{for\ i\ =\ 0\ to\ 255}$

1.1 $\boldsymbol{set\ V[i] = 0}$

2. $\boldsymbol{for\ i\ =\ 0\ to\ row * col}$

2.1 $\boldsymbol{set\ V[Image[i]]\ =\ V[Image[i]] + 1}$

3. $\boldsymbol{for\ i\ =\ 0\ to\ 255}$

3.1 $\boldsymbol{set\ V[i] = {V[i]}/{row * col}}$

3.4.2 Stochastic Learning Weak Estimator

The Stochastic Learning Weak Estimator, or SLWE, was proposed by Ommen et al. [46] 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.

SLWE was developed using stochastic learning. For each estimate performed, each instant is updated, based on the values of the current sample. This updating occurs by using a multiplication rule.

Let X be a binomially distributed random variable, which takes on the value of either "1" or "2". We assume that X obeys the distribution S, where $S = [s_1, s_2]^T$. In other words,

X = "1" with probability s_1

= "2" with probability s_2,

where, $s_1 + s_2 = 1$.

Let $x(n)$ be a concrete realization of X at time "n", then S can be estimated by maintaining a running estimate of $P(n) = [p_1(n), p_2(n)]^{\mathrm{T}}$ of S, where $p_i(n)$ is the estimate of s_i at time "n", for i = 1, 2. The value of $p_1(n)$ is updated as per the following simple rule:

$$p_1(n+1) = \begin{cases} \lambda p_1(n), & if\ x[n] = 2 \\ 1 - \lambda p_1(n), & if\ x[n] = 1 \end{cases}$$

Where λ is user-defined parameter $0 < \lambda < 1$ and

$$p_2(n+1) = 1 - p_1(n+1)$$

Let X be a multinomial distributed random variable, which takes on the values from the set {"1" , . . . , " r" }. We assume that X is governed by the distribution

$S = [s_1, \ldots, s_r]^{\mathrm{T}}$ as follows:

X = "i" with probability s_i , where $\sum_{i=1}^{r} s_i = 1$.

Now, in our model we let $x(n)$ be a concrete realization of X at time "n", where $X = \{0, \ldots, 255\}$. The main aim is to estimate S, i.e., s_i for $i = 0, \ldots, 255$. We achieve this by maintaining a running estimate $P(n) = [p_0(n), \ldots, p_{255}(n)]^{\mathrm{T}}$ of S, where $p_i(n)$ is the estimate of s_i at time "n", for $i = 1, \ldots, r$. Then, the value of $p_i(n)$ is updated as per the following simple rule:

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

where $i = 0, \ldots, 256$ and λ is training constant. [25 and 47] reported that the value of λ was an estimated ~0.999. From our experiments, it was found that the optimal value λ was ~0.9995.

Further more [46] show that when $n \rightarrow \infty$

$$E[V_i(\infty)] = s_i$$

Thus, $\sum_{i=0}^{255} V_i\,(n) \approx 1$.

The feature vector for the three colour components are calculated in the same manner as mentioned above. Each colour component of the image is transformed into a one dimensional array. Algorithm 3.3 describes the process of calculating the feature vector and Algorithm 3.4 implements the SLWE sum rule.

Algorithm 3.3: $\boldsymbol{SLWE(Image*, row, col, \lambda)}$

---

1. **for** $\boldsymbol{i = 0\ to\ n}$

1.1 $\boldsymbol{set\ V[i] = 1 / 256}$

2. **for** $\boldsymbol{i = 0\ to\ row * col}$

2.1 **set** $\boldsymbol{temp = SLEW\_sum(V,\ Image[i], \lambda)}$

2.2 **for** $\boldsymbol{j = 0\ to\ 255}$

2.2.1 **set** $\boldsymbol{V[j] = \lambda * V[j]}$

2.3 **set** $\boldsymbol{V[Image[i]] = temp}$

$Image*$ is converted into a one dimensional array.

Algorithm 3.4: $\boldsymbol{SLWE\_sum(V, n, \lambda)}$

---

1. **set** $\boldsymbol{sum = 0}$
2. **for** $\boldsymbol{i = 0\ to\ 255}$

2.1 **set** $\boldsymbol{sum = sum + V[i]}$

3. **set** $\boldsymbol{sum = sum - V[n]}$
4. **set** $\boldsymbol{sum = V[n] + (1 - \lambda) * sum}$

3.5 Classification Distances

Statistical distance measure, defined as the distances between two probability distributions is used in many research areas in pattern recognition, information theory, and communication. It captures correlations or variations between attributes of the feature vectors and provides bounds for probability of retrieval error of a two way classification problem.

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 [34-36]. This list includes the Euclidean distance (ED), the Weighted Euclidian distance (WED) and Non-negative Similarity Coefficient-based distance (NVSC). Figure 3-11 shows image classification process.

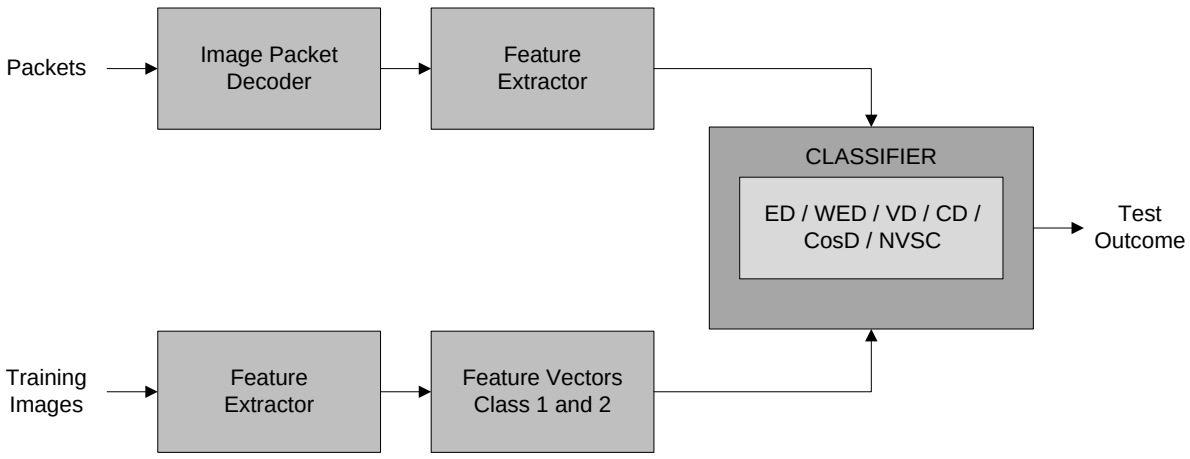


Figure 3- 11: Block diagram showing classification process

3.5.1 Euclidian Distance

In practice, for a simple pattern recognition problem, Euclidian distance (ED) is considered the simplest way to quickly measure the dissimilarity between two histograms at a low computational cost. For feature vectors with 256 symbols $V = (v_0, \dots, v_{255})$ and $V' = (v_0', \dots, v_{255}')$, the ED between these vectors can be calculated as:

$$d(V, V') = \sqrt{\sum_{i=0}^{255} (v_i - v_i')^2}$$

The problem with ED is that the relationship between the entries of feature vectors is discarded, and as a result, it does not reflect the true distance between feature vectors V and V'.

Algorithm 3.5: $\boldsymbol{ED}\ (\boldsymbol{V}, \boldsymbol{V'})$

---

1. **set** $\boldsymbol{d} = \mathbf{0}$
2. **for** $\boldsymbol{i} = \mathbf{0}\ \boldsymbol{to}\ \mathbf{255}$

2.1 **set** $\boldsymbol{d} = \boldsymbol{d} + (\boldsymbol{V}[\boldsymbol{i}] - \boldsymbol{V'}[\boldsymbol{i}])^2$

3. **set** $\boldsymbol{d} = \sqrt{\boldsymbol{d}}$

3.5.2 Weighted Euclidian Distance

Weighted Euclidian Distance (WED) is based on correlations between feature vectors by which different patterns can be identified and analyzed. It is a useful way of classifying unknown sample sets into known sample sets. It differs from Euclidean distance in that it takes into account the correlations of the data set and is scale-invariant, i.e. not dependent on the scale of measurements. WED between two vectors can be calculated as:

$$d(V, V') = \sqrt{\sum_{i=0}^{255} \frac{(v_i - v_i')^2}{\sigma_i^2}}$$

Where σ_i is the standard deviation of v_i over the training set, and defined as:

$$\sigma_i = \sqrt{\frac{1}{N}\sum_{j=1}^{N}(v_{ij} - \bar{V}_i)^2}$$

Where $\bar{V}_i$ is the mean of V_i, and defined as:

$$\bar{V}_i = \frac{1}{N}\sum_{j=1}^{N} v_{ij}$$

Algorithm 3.6: $\boldsymbol{WED}$ $(\boldsymbol{V}, \boldsymbol{V'})$

---

1. **set** $\boldsymbol{d} = \mathbf{0}$
2. **for** $\boldsymbol{i} = \mathbf{0}$ ***to*** $\mathbf{255}$

2.2 **set** $\boldsymbol{d} = \boldsymbol{d} + (\boldsymbol{V}[\boldsymbol{i}] - \boldsymbol{V'}[\boldsymbol{i}])^2 / \boldsymbol{var}(\boldsymbol{v_i})$

3. **set** $\boldsymbol{d} = \sqrt{\boldsymbol{d}}$

3.5.3 Variational Distance

For feature vectors with 256 symbols $V = (v_0, \ldots, v_{255})$ and $V' = (v_0', \ldots, v_{255}')$, the Variational Distance (VD) between these vectors can be calculated as:

$$d(V, V') = \sqrt{\sum_{i=0}^{255} |v_i - v_i'|}$$

Algorithm 3.7: $\boldsymbol{VD}$ $(\boldsymbol{V}, \boldsymbol{V'})$

1. **set** $\boldsymbol{d} = \boldsymbol{0}$
2. **for** $\boldsymbol{i} = \boldsymbol{0}$ $\boldsymbol{to}$ $\boldsymbol{255}$

2.3 **set** $\boldsymbol{d} = \boldsymbol{d} + \boldsymbol{abs}(\boldsymbol{V}[\boldsymbol{i}] - \boldsymbol{V'}[\boldsymbol{i}])$

3. **set** $\boldsymbol{d} = \sqrt{\boldsymbol{d}}$

3.5.4 Counter Distance

Counter distance is another simple distance measure designed to calculate the distance as follows:

$$d_1 = \#\ elements\ such\ that$$

$$|v_{1i} - v_i^{'}| < |v_{2i} - v_i^{'}|$$

$$d_2 = \#\ elements\ such\ that$$

$$|v_{1i} - v_i^{'}| \geq |v_{2i} - v_i^{'}|$$

Algorithm 3.8: $\boldsymbol{CD}$ $(\boldsymbol{V1}, \boldsymbol{V2}, \boldsymbol{V'})$

1. set $\boldsymbol{d1} = \boldsymbol{0}$
2. set $\boldsymbol{d2} = \boldsymbol{0}$
3. for $\boldsymbol{i} = \boldsymbol{0}$ $\boldsymbol{to}$ $\boldsymbol{255}$

3.1 **if** $\boldsymbol{abs}(\boldsymbol{V1}[\boldsymbol{i}] - \boldsymbol{V'}[\boldsymbol{i}]) < abs(V\boldsymbol{1}[\boldsymbol{i}] - \boldsymbol{V'}[\boldsymbol{i}])$

3.1.1 $\boldsymbol{set}\ \boldsymbol{d1} = \boldsymbol{d1} + \boldsymbol{1}$

3.2 **else**

3.1.2 $\boldsymbol{set}\ \boldsymbol{d2} = \boldsymbol{d2} + \boldsymbol{1}$

3.5.5 Cosine Distance

Cosine Distance (CosD) is also another similarity measure that is very similar to ED in higher dimensional spaces [34]. Qian [35] reported that among all the conventional distance measures like ED, the cosine distance achieved the best result in facial recognition tests and was ranked as one of the best 3 measures out of 17 different distance measures. CosD between two vectors can be calculated as:

$$d(V, V') = 1 - \frac{\sum_{i=0}^{255} v_i v_i'}{\sqrt{\sum_{i=0}^{255} {v_i}^2} \quad \sqrt{\sum_{i=0}^{255} {v_i'}^2}}$$

Algorithm 3.9: $\boldsymbol{CosD}$ ($\boldsymbol{V}, \boldsymbol{V'}$)

1. **set** $\boldsymbol{d = 0}$
2. **set** $\boldsymbol{temp1 = 0}$
3. **set** $\boldsymbol{temp2 = 0}$
4. **for** $\boldsymbol{i = 0}$ ***to*** $\boldsymbol{255}$

4.1 **set** $\boldsymbol{d = d + V[i] * V'[i]}$

4.2 **set** $\boldsymbol{temp1 = V[i]^2}$

4.3 **set** $\boldsymbol{temp2 = V'[i]^2}$

5. **set** $\boldsymbol{d = 1 - d / \left(\sqrt{temp1} * \sqrt{temp2}\right)}$

3.5.6 Non-negative Vector Similarity Coefficient-based Distance

Non-negative vector similarity coefficient-based (NVSC) distance is our last distance measure which originated from the theory of multivariate clustering analysis, but has

showed satisfactory results in face recognition [36]. NVSC is derived from a similarity coefficient specifically for non-negative vectors and defined as:

$$d(V, V') = \frac{\sum_{i=0}^{255} min(v_i, v_i')}{\sum_{i=0}^{255} max(v_i, v_i')}$$

Algorithm 3.9: $\boldsymbol{NVSC}$ $(\boldsymbol{V}, \boldsymbol{V'})$

---

1. **set** $\boldsymbol{d = 0}$
2. **set** $\boldsymbol{temp1 = 0}$
3. **set** $\boldsymbol{temp2 = 0}$
4. **for** $\boldsymbol{i = 0}$ $\boldsymbol{to}$ $\boldsymbol{255}$

4.1 **set** $\boldsymbol{temp1 = temp1 + min(V[i], V'[i])}$

4.2 **set** $\boldsymbol{temp2 = temp2 + max(V[i], V'[i])}$

5. **set** $\boldsymbol{d = temp1/temp2}$

3.6 Image Matching

Image matching is another useful host-based application of our approach that is applied to match black-listed images in the network. This system does not rely upon previously encountered imaging data and it can be applied against images which have been manipulated. This eliminates the weakness exhibited by most hash-based matching algorithms which rely upon consistent data for accuracy. Hash-algorithms are prone to failure when any changes in a given image (even one pixel) are presented, thus resulting in a mismatch. Illicit images on the internet are often times stored and displayed as thumbnails, resized, watermarked, cropped, and altered in many ways.

To perform the image matching process, feature vectors of target images are extracted and stored in a database using MLE or SLWE. Every image passing through the system is first calculated by its feature vector using MLE or SLWE, depending on which database is used, then, it is compared against a database as illustrated in Figure 3-12. If the distance is within a given threshold then a result of a match is recorded. If the threshold is not reached, the result is recorded as a non-match. Our experiments show that the threshold rates for accurate matching occur at 0.0025 for ED and 0.34 for VD. The image matching system is described in Algorithm 3.10.

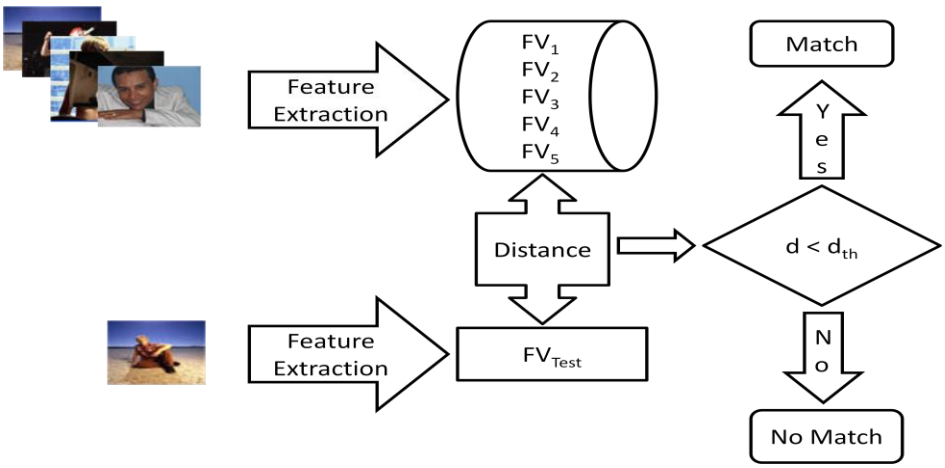


Figure 3- 12: Image matching system

Algorithm 3.10: $\boldsymbol{ImageMatch(Image, row, col, *V)}$

1. **set** $\boldsymbol{V' = MLEestimator(Image, row, col)}$
2. **for** $\boldsymbol{i = 1\ to\ number\_of\_images}$

2.1 **set** $\boldsymbol{d = * classifier(V, V')}$

2.2 **if** $\boldsymbol{d <= * match_{threshold}}$ then match found

$* classifier$ could be any of the distance measures(ED, WED, VD, CD, CosD, NVSC)

$* V$is feature vectors of black-listed images, and $V[i]$ is feature vector for image i.

$* match_{threshold}$ is different for all distance measures.

Chapter 4

RESULTS AND DISCUSSION

The experiments for this thesis 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. Our experiment was divided into three categories: image classification, image matching and network performance. The network environment framework is illustrated in Figure 4-1.

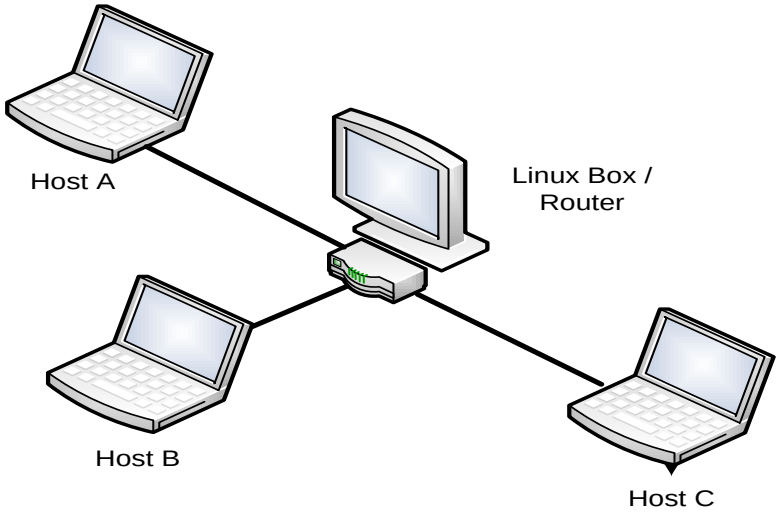


Figure 4- 1: Experimental setup

4.1 Image Classification

Image classification is an important part of digital image analysis. This classification stage is divided into two stages – training and classification, and is done for both compressed binary data and RGB data. A further explanation is detailed in the upcoming sections.

4.1. 1 Training Stage

All images used in our experiments were legal. No actual child pornography images were ever collected, seen, sought, or used.

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. The first method was training by using a compressed byte stream. In the second method of training, the system was trained with an RGB colour space in total, and then each RGB colour component was individually trained. For the RGB colour space and compressed byte stream training, the feature vectors of individual components were extracted using MLE and SLWE algorithms as discussed in Chapter 3.

4.1. 2 Classification Stage

Every image that flowed through our network was intercepted and tested for its content. For the first stage of our testing, the feature vector for intercepted images was calculated from the compressed byte stream data and compared with feature vectors of nude and non-nude image vectors derived from the training stage. Table 4-1 shows that the success rate for byte stream data. Classification distance algorithm, 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.

Training Algorithms	**Success Rate (%)**					
	ED	**WED**	**VD**	**CD**	**CosD**	**NVSC**
MLE	63.6	65.4	68.3	61.7	66.5	67.9
SLWE	67.5	66.3	69.4	60.3	64.3	69.1

Table 4-1: Success rate for nude images using JPEG byte stream data

Image classification for RGB colour space is done by using a "height" scan and a "radius" scan (see Figure 4-2) with a combination of RGB components or by using individual Red, Green or Blue components.

Height scan and radius scans serve two primary reasons for their use. Firstly, 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. Secondly, it was observed that system performance will increase if features which are determined to have no effect on image analysis are removed.

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%. This was then tested with 3000 images for classification using six distance measures algorithms as explained in Chapter 3. Half of these tested images were part of the nude content category, with sizes similar to images used in the training stage. For RGB colour space each image was processed using a height scan and radius scan method.

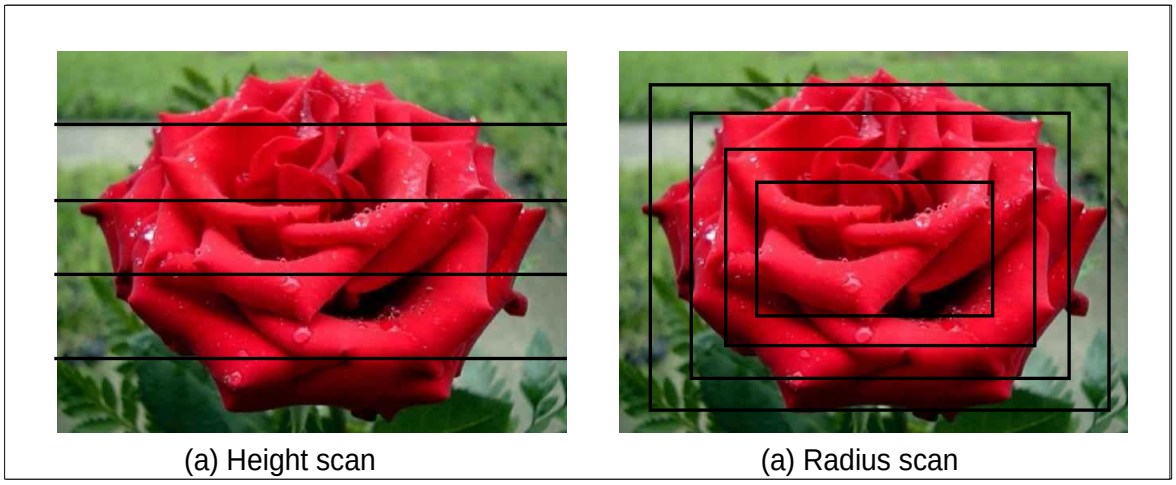


Figure 4- 2: Height and radius scan methods

Figure 4-3 shows the success rate of the MLE engine when the height scan method was used for images with nude content. CosD, NVSC and WED yielded better results compared to the CD, VD and ED algorithms. Results from this test also indicate that CosD, NVSC and WED algorithms perform similarly when processing 50% of the image and when processing 90% of the image area processed in the other three algorithms.

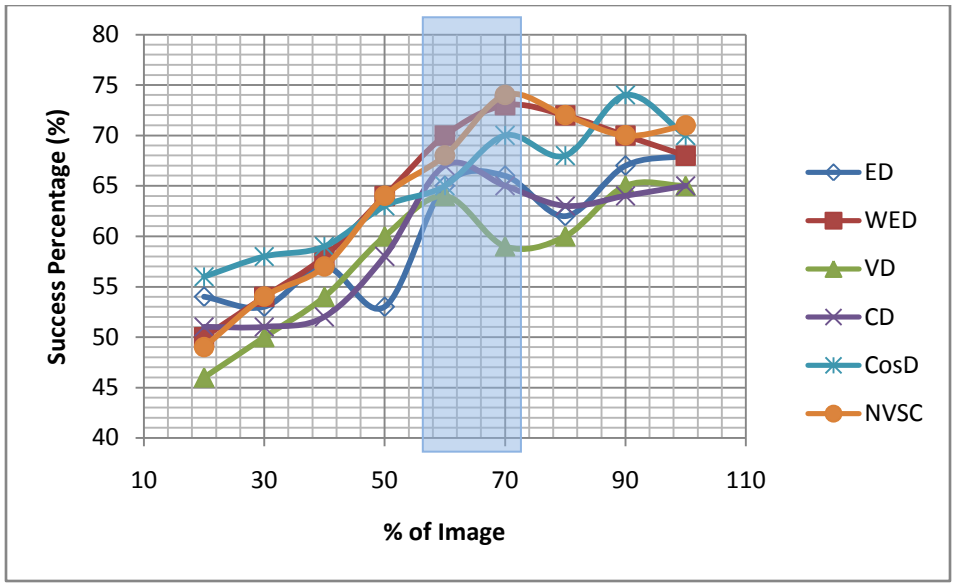


Figure 4- 3: Success rate for nude images using height scan (MLE).

Also, it was observed that the optimal percentage to scan was 65-80% of the area of the test image, as most of the characteristic features of these images are located within the top 65-80% of the image for height scan (c.f shadow area at Figure 4-3).

This result is more accurate for images processed with the radius scan method. Figure 4-4 below shows success percentages increasing to as much as 75% when the radius scan method was used with images containing nudity. Also, it was observed that the optimal percentage to scan was 70-80% of the area of the test image.

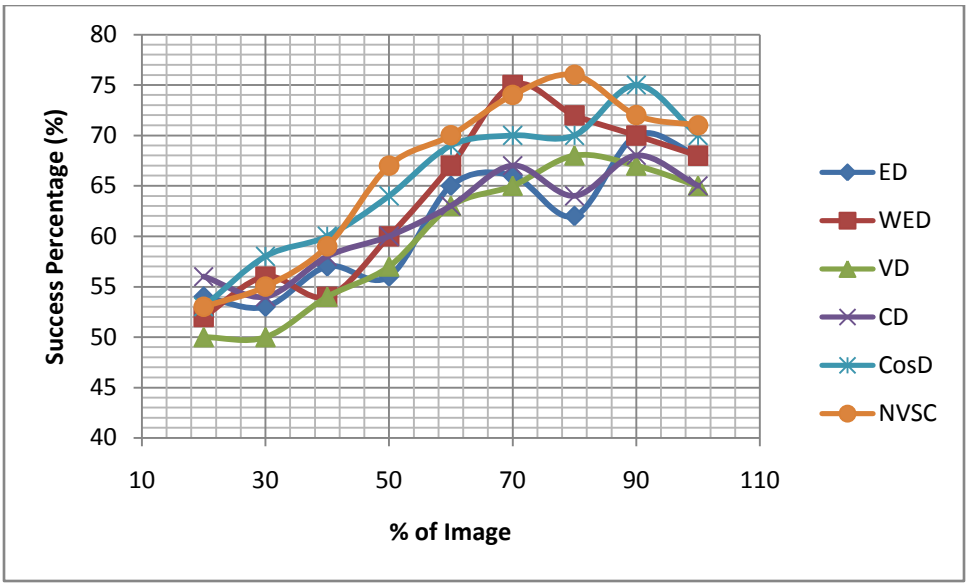


Figure 4- 4: Success rate for nude images using radius scan (MLE).

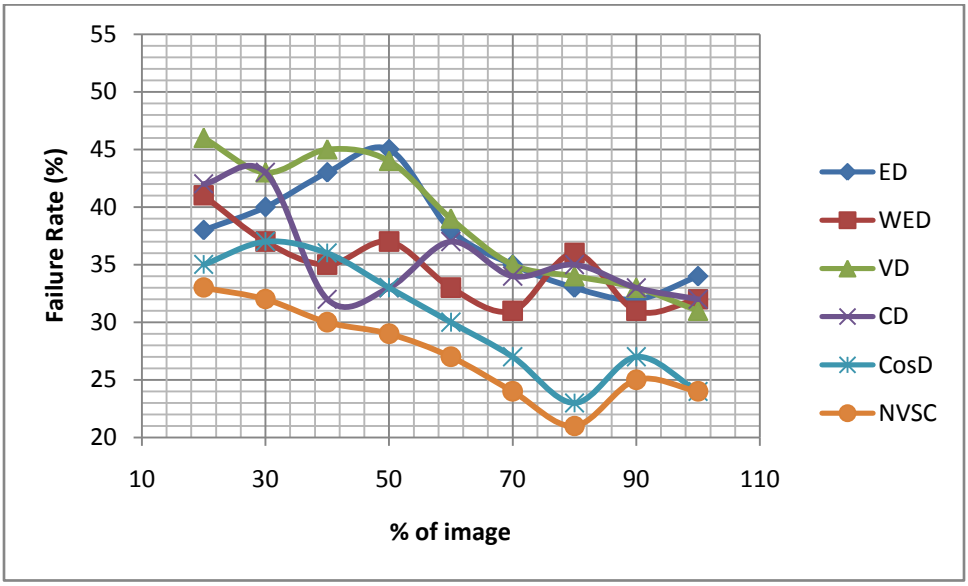


Figure 4- 5: Error rate using height scan; all test images are non-nude (MLE).

Figure 4-5 shows that the system falsely identified non-nude images as nude images in the MLE engine when the height scan method was used. CosD and NVSC demonstrated better results as compared to the WED, CD, VD and ED algorithms. These false positives were immensely reduced when the scanning area of the image was increased.

Also, Figure 4.6 shows results from the radius scan method which provided comparable results to the height scan method, with a minor decrease in error percentage rate.

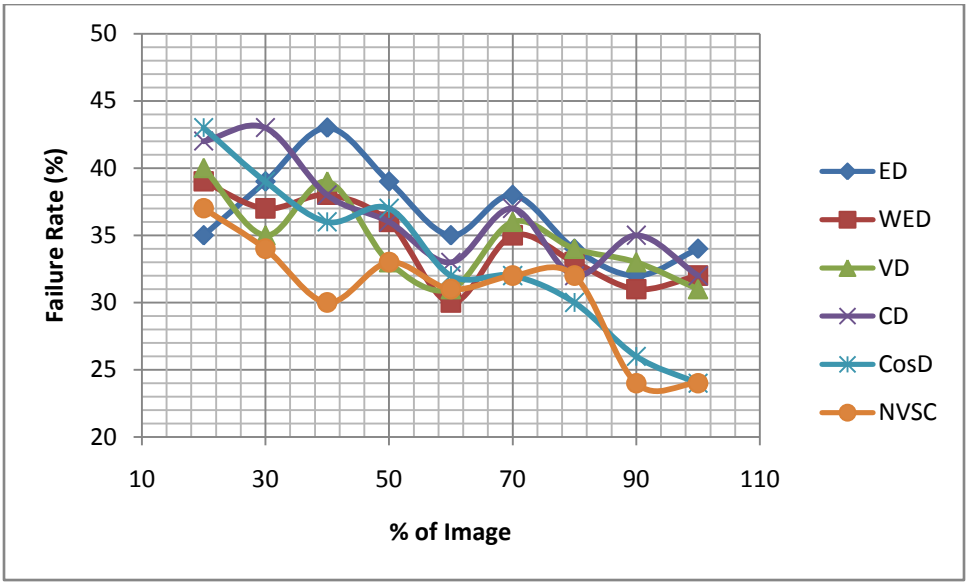


Figure 4- 6: Error rate using radius scan; all test images are non-nude (MLE).

Figure 4-7 & 4-8 show that the success rate of the SLWE algorithm yield better results with the VD algorithm. Moreover, the radius scan method still provided better performance over the height scan method.

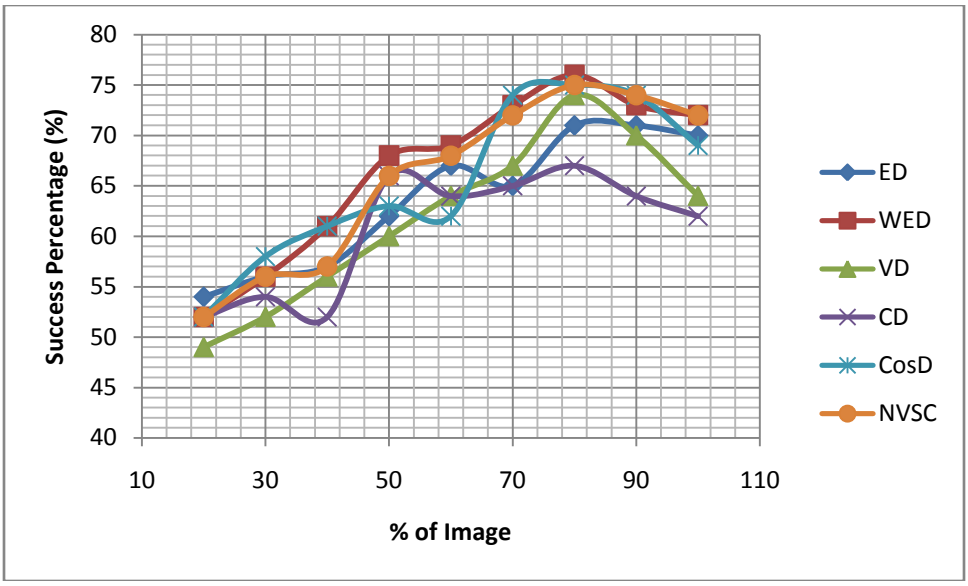


Figure 4- 7: Success rate for nude images using height scan (SLWE).

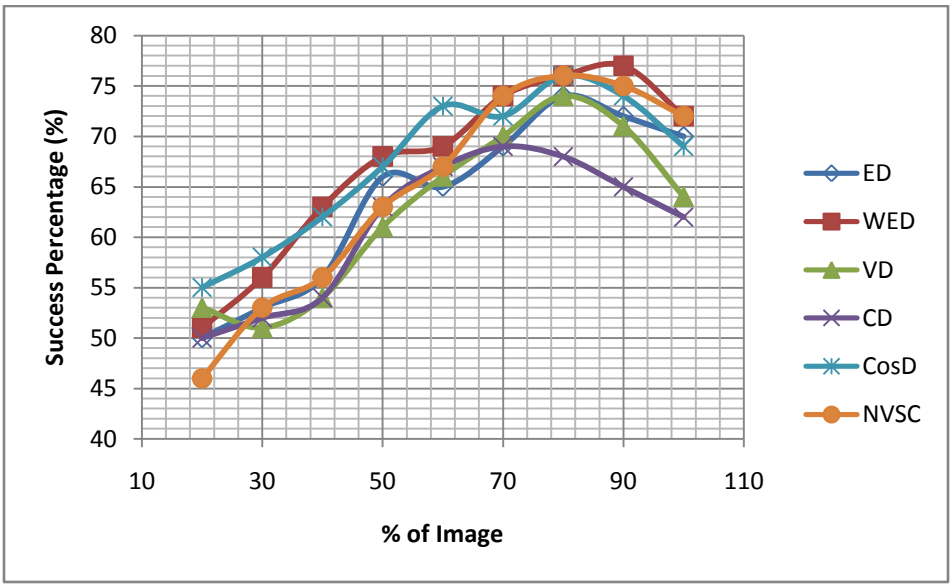


Figure 4- 8: Success rate for nude images using radius scan (SLWE).

Overall, the SLWE algorithm improved the system performance by ~2 – 4%. Other important observation was that when the individual colour components were compared, the success rate was not distant from its RGB counterpart. Refer to Appendix C for

individual colour components accuracy test results. Tables 4-2 and 4-3 illustrate overall performance of MLE and SLWE algorithms when using height and radius scan.

Algorithm	Best	Worst	Observation
MLE	NVSC with 73.5 % success rate	CD with 55.8 % success rate	Similar results are observed when considering only the B component of RGB colour space.
SLWE	WED with 76.4 success rate	CD with 64.3 % success rate	NVSC and CosD have similar best result

Table 4-2: Best/worst performance for MLE and SLWE algorithms (Height scan, when considering 70-80% the image)

Algorithm	Best	Worst	Observation
MLE	NVSC with 76 % success rate	ED with 60.8 % success rate	Similar results are observed when considering only the B component of RGB colour space.
SLWE	WED with 77% success rate	CD with 66.3 % success rate	NVSC and CosD have similar best result

Table 4-3: Best/worst performance for MLE and SLWE algorithms (Radius scan, when considering 70-80% the image)

4.2 Image Matching

For matching, the image database was composed of feature vectors of 20 different images randomly taken which included nude, semi-nude and non-nude. An initial pool of 20 images were altered from their originals in size, texture and with various random effects, creating a total of 55 images exactly related to the original 20 in content.

We also added 200 extra images to our test data. We found that the system identified 95% of those images in the database and there were no false positives. The system was unable to identify the other 5% of the images as black listed images. In the future, a further refined threshold can help reduce the number of errors in the system. Figure 4-9 shows, on the left, sample images in the database and, on the right, the images matched with the database. The first image set (A) modified images to a variety of sizes. The second image set (B) was resized and shaded. The third image set (C) was modified in colour and resized. The fourth image set (D) was tested with a combination of blur and sizes, and finally, the fifth image set (E) was shaded and sized.

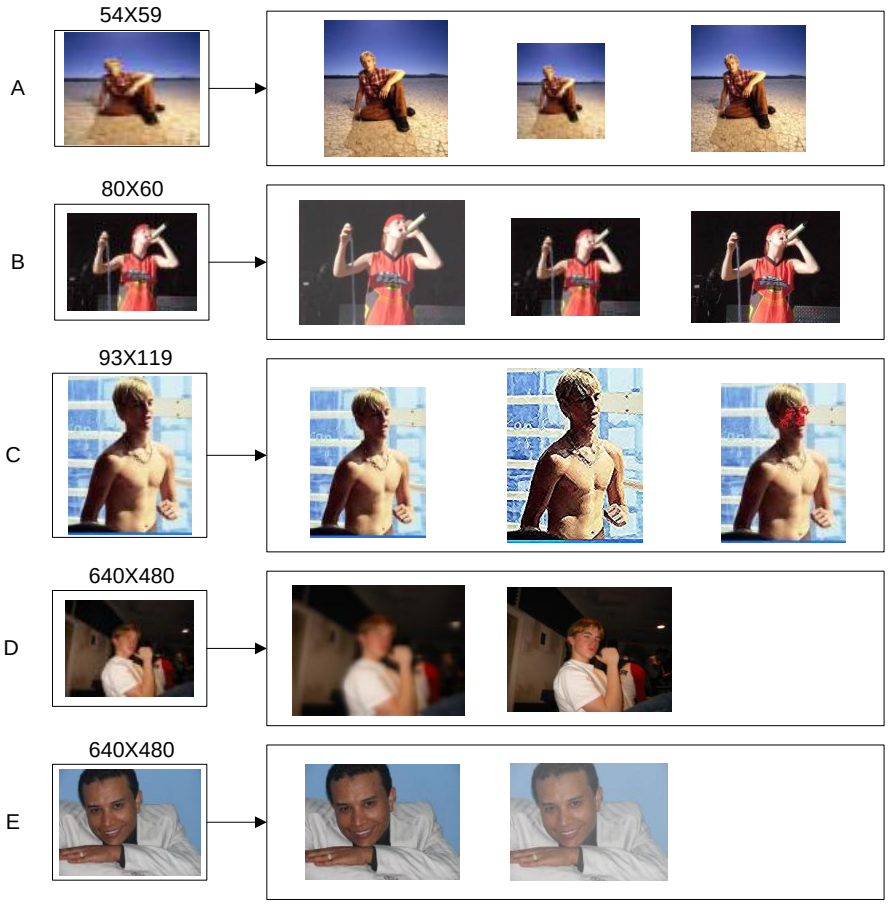


Figure 4- 9: Results from image matching algorithms

4.3 Network Performance

Using Wireshark [40] in capture mode, network performance was measured while the first 100 images were transferred between host *A* and host *B*. A time measurement was performed between filtered and unfiltered processing to determine the amount of overhead generated by filtering.

This filter was enabled in various modes such as byte stream filter mode using MLE and SLWE, RGB filtering modes for both MLE and SLWE and for one colour component, also using MLE and SLWE.

As a result of these experiments, tests showed the best timings were achieved using byte stream with MLE. 11% to 17% of overhead occurred when using ED and WED, respectively. Using SLWE for byte stream, results as high as 80% to as low as 58% occurred. Figure 4-10 and 4-11 show the byte stream processing times and the overhead as a percentage for MLE and SLWE. Refer to Appendix B for more network performance test results.

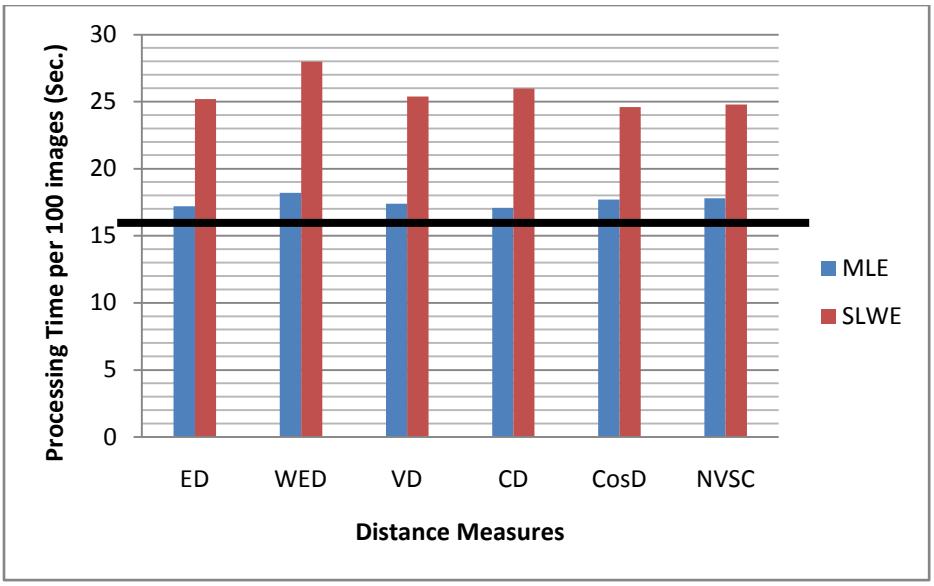


Figure 4- 10: Processing times of 100 images when byte stream used; black line shows processing time without filter enabled.

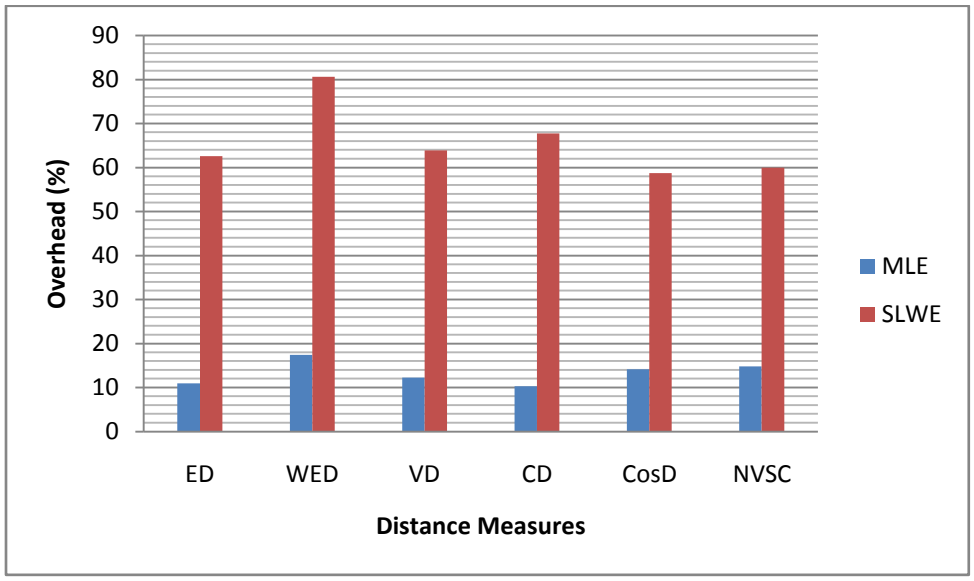


Figure 4- 11: Percent overhead when considering image byte stream

Chapter 5

CONCLUSIONS AND RECOMMENDATIONS

Detection and prevention of electronically transmitted illicit pornography involves complex technical measures in relation to still images and moving video. No approach thus far has generated a truly accurate system to fully mitigate the electronic transmission of child pornography. However, this thesis is an important milestone in the ongoing effort to effectively produce a total solution within the privacy framework in effect in Canada. All images used in our experiments were legal. No actual child pornography images were ever collected, seen, sought, used or considered.

5.1 Summary and Conclusions

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. Of these two methods, it was determined through our experimentations that various methods proved more accurate and efficient than others, but were dependent on the exact method they were to be used. For example, our experiments showed that the SLWE coupled with the NVSC distance algorithm was highly accurate for the RGB colour 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, MLE and VD algorithms 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 this system unusable in any practical application. Algorithms which showed promising results in terms of accuracy and success or error rates, such as SLWE coupled with linear classifiers such as WED, ND, and NVSC, are unlikely to be used in network-based implementations, as the imposed overhead is much higher. SLWE is roughly 3 to 4 times higher in network overhead than MLE. The SLWE algorithm takes 255 steps more than that of MLE to update its feature vectors in every step although MLE and SLWE algorithms have the computational complexity of $O(n)$, where n is the size of the image.

When a comparison is made between the data types used for training and classification the RGB colour space had more favorable performance than that of byte stream by as much as 7 percent. Moreover it was observed that similar results were obtained if only one colour component is used thereby decreasing the processing times by factor of 3.

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.

A system with little overhead is required for the efficient operation of our application at the network layer. To accomplish this, MLE coupled with variational distance using a compressed byte stream was determined to be the best choice for the classification of illicit images at the network layer. Advantages of this method are that only 12 percent of overhead is generated with an average of 71% success rate when classifying images at the network layer in real-time. If time based performance is a non-issue, then SLWE coupled with NVSC or WED would provide a 76% success rate.

Furthermore, 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 our thesis 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.

5.2 Recommendations and Future Work

The primary purpose of network layer image classification is to classify images with high success rates and minimal overhead. To satisfy both constraints is a difficult challenge, yet, it may be possible to develop a system with higher success rates and lower overhead in the near future.

A noticeable quality of our system remains the decision making functionality which allows nude or non-nude detection. There were multiple instances of images with great distance divides between both feature vectors and yet the system still identified the images as nude. To resolve this issue, an experiment may be conducted to improve thresholds for those image distances well above a defined limit. An additional class with three class labels instead of two (neither nude or non-nude, nude, and non-nude) would help to decrease error rates.

A particular recommendation is to use Che-Jen's et al. [29] approach, which uses a digital feature retrieval mechanism for JPEG-based images using only the DC coefficients of the image. Our system could be trained using DC coefficients, and it extracts the JPEG binary data at the network layer for classification. Instead of decoding the entire byte stream, it uses JPEG image markers to extract only the DC values and then applies them to the classification algorithm. This method should work to considerably minimize the time required to classify a given set of images.

For proceeding with research in this field, a system should be first created using a text-based method to scan and analyze real-time data traffic form blacklisted URLs, image meta-data, filenames and keywords. Should the results of the scan prove positive, the image classification system should not be engaged. If the results of this scan do not show any conclusive results, then the image classification system should be engaged for further analysis of that data-traffic.

Another approach is to use a statistical method to estimate the number of illicit images in a sample set of data traffic. This feature should trigger the image classification system in a randomly distributed time interval over an increasing amount of time to statistically estimate ratio of illicit images to non-illicit images for the current data traffic. This would eventually reduce the overhead since the system would not scan all the data traffic.

REFERENCES

[1] R. F. Jørgensen. "Blocking Access to Child Pornography - The Danish Case," 2007, Available online: http://ec.europa.eu/information_society/activities/sip/docs/forum_june_2006/rikke_frank_joergenseng.pdf [Last Accessed: 14th Feb. 2009].

[2] E. Grochowski and R. D. Halem. "Technological Impact of Magnetic Hard Disk Drives on Storage Systems," *IBM Systems Journal*, Vol. 2, pp. 338-346, 2003.

[3] E. Lie and T. Reynolds, "Birth of Broadband - ITU Internet Reports," Fifth ed. 2003

[4] A. Alikhan, T. W. Luckinbill, S. S. Lee and E. S. McFadden, "Report of the Department of Justice's Task Force on Intellectual Property," *U.S. Department of Justice, Office of the Attorney General*, 2004, Available online: http://www.usdoj.gov/criminal/cybercrime/IPTaskForceReport.pdf .

[5] T. Frieden, "27 charged in child porn sting," *CNN.com*, March16, 2006. Available online: http://www.cnn.com/2006/LAW/03/15/childporn.arrests/index.html [Last Accessed: 12th Aug. 2008].

[6] Toronto Police, "Child Exploitation Tracking System (CETS) Fact Sheet," 2006, Available online: http://www.torontopolice.on.ca/media/text/20060831-cetsfactsheet.pdf [Last Accessed: 9th Feb. 2007].

[7] B. Warner, "Police to Launch International Cyber Child Porn Sting," 2003, Available online: http://www.castlecops.com/a4498Police_to_Launch_International_Cyber_Child_Porn_Sting.html, [Last Accessed: 27th Feb. 2007].

[8] J. Kevin, "FBI Arrests 40 in Child Porn Sting," 2002 Available online: http://www.usatoday.com/tech/news/2002/03/18/net-porn.htm, [Last Accessed: 20th Aug. 2007]

[9] Privacy International, "TTC Privacy Complaint Letter," P*rivacy International, London, England* 2007, Available online: http://www.privacyinternational.org/issues/compliance/complaint_ttc_privacy.pdf.

[10] A. Cavoukian, "Privacy and Video Surveillance in Mass Transit Systems: A Special Investigation Report," *Privacy Investigation Report MC07-68*, 2008, Available online: http://www.ipc.on.ca/images/Findings/mc07-68-ttc_592396093750.pdf.

[11] "ATM Safety Act," June 1995, Available online: http://www.banking.state.ny.us/legal/atmsafe.htm.

[12] "Communication Assistance for Law Enforcement Act, CALEA," *U.S.C 1002,* pp. 103, Sept. 2005, Available online: http://www.fcc.gov/calea/.

[13] Lee., "The Porn Breakers," *The engineer 291(7610),* pp. 30. 2002

[14] X. Fan, X. Xie, Z. Li, M. Li and W. Ma, "Photo-to-Search: Using Multimodal Queries to Search the Web from Mobile Devices," *Proceedings of the 7th ACM SIGMM International Workshop on Multimedia Information Retrieval*, pp. 143-150, 2005.

[15] P. J. Resnick, D. L. Hansen and C. R. Richardson, "Calculating Error Rates for Filtering Software," *Communication of ACM 47(9),* pp. 67-71, 2004. Available online: http://doi.acm.org/10.1145/1015864.1015865

[16] Net Nanny Inc. Available online: http://www.netnanny.com/, [Last Accessed: 18th Aug. 2007].

[17] Cyber Patrol Parental Control Software, Available online: http://www.cyberpatrol.com/, [Last Accessed: 27th Aug. 2007].

[18] P. Gupta and N. McKeown, "Algorithms for Packet Classification," *Computer Systems Laboratory, Stanford University,* 2001.

[19] A. Feldman and S. Muthukrishnan, "Tradeoffs for Packet Classification," Vol. 3, pp. 1193 - 1202, March 26-30, 2000.

[20] Y. Wang, Y. Zhang, T. Tang, A. Krishnamurthy, G. Damm and B. Bou-Diab, "Novel Disjoint Graph Based Algorithm for Multi-field Range-based Packet Classification," *2004 IEEE International Conference on Communications,* vol. 2, pp. 1108 - 1112, June 20 - 24. 2004.

[21] P. Warkhede, S. Suri and G. Varghese, "Fast Packet Classification for Two-Dimensional Conflict-Free Filters," *Proceedings of the Twentieth Annual Joint Conference of the IEEE Computer and Communications Societies,* Vol. 3, April 22-26. 2001.

[22] S. Dharmapurikar, H. Song, J. Turner and J. Lockwood, "Fast Packet Classification using Bloom Filters," Presented at ANCS '06: *Proceedings of the 2006 ACM/IEEE Symposium on Architecture for Networking and Communications Systems,* Available online: http://doi.acm.org/10.1145/1185347.1185356

[23] F. Yu, T. V. Lakshman, M. A. Motoyama and R. H. Katz, "Efficient Multimatch Packet Classification for Network Security Applications," *Selected Areas in Communications, IEEE Journal on,* vol. 24, pp. 1805 - 1816, Oct. 2006.

[24] K. Wang and S. J. Stolfo, "Anomalous Payload-Based Network Intrusion Detection," *International Symposium on Recent Advances in Intrusion Detection,* Sept. 15-17. 2004.

[25] A. Shupo, M. V. Martin, L. Rueda, A. Bulkan, Y. Chen and P. C. K. Hung, "Toward Efficient Detection of Child Pornography in the Network Infrastructure," *IADIS International Journal on Computer Science and Information Systems,* vol. 1, pp. 15-31, 2006.

[26] C. Munish, M. V. Martin, L. Rueda and P. C. K. Hung, "Toward New Paradigms to Combating Internet Child Pornography," *Electrical and Computer Engineering, 2006. CCECE '06. Canadian Conference,* pp. 1012-1015, May. 2006.

[27] M. Chopra, M. V. Martin, L. Rueda and P. C. K. Hung, "A Source Address Reputation System to Combating Child Pornography at the Network Level," *IADIS Internl. Conf. on Applied Computing,* Feb. 2006.

[28] Websense Enterprise, "Employee Internet Management Software Solution," *Cisco Pix Edition* 2002, Available online: http://www.uit.co.uk/wbs/pdf/web_ciscopix.pdf

[29] H. Che-Jen, L. Wei-Cheng, L. Jung-Shian, "An Efficient Packet-level JPEG Forensic Data Collection," *Future generation communication and networking,* Vol. 2, pp.108 – 113, Dec. 2007.

[30] M. P. Ryan, A. D. Whitehead, "Method and Device for Classifying Internet Objects and Objects Stored on Computer-readable Media," Document No: 7383282, Application No: 09/978,182, Application Date: Oct. 17, 2001.

[31] L. J. Hove, "Extending Image Retrieval Systems with a Thesaurus for Shapes," Masters Thesis, *Institute for Information and Media Sciences*, University of Bergen, April 2004.

[32] Z. Feng, D. Tien, "Enhancement of Semantics in CBIR," *Third International Conference on Information Technology and Applications, 2005, ICITA 2005,* Vol.1, pp. 744-745, July 2005.

[33] "Personal Information Protection and Electronic Document Act, PIPEDA", *Statues of Canada 2000, Chapter 5,* Sept. 2000, Available online: http://www.privcom.gc.ca/legislation/02_06_01_01_e.asp.

[34] Association Xpertise Inc., "The PIPEDA Privacy Principles, A Guide for Associations and Non-Profit organization," Calgary 2001, Available online: www.axi.ca.

[35] G. Qian, S. Sural, Y. Gu, and Pramanik, "Similarity Between Euclidean and Cosine Angle Distance for Nearest Neighbor Queries," *In Proceedings of the 2004 ACM Symposium on Applied Computing,* ACM, New York, NY, pp.1232-1237, March 2004.

[36] Y. Xue, C. S. Tong, W. Zhang, "Evaluation of Distance Measures for NMF-based Face Recognition," *International Conference on Computational Intelligence and Security*, Vol. 1, pp. 651 – 656, Nov. 2006.

[37] Joint Photographic Experts Group, Available online: http://www.gpeg.org/, [Last Accessed: 1st March, 2009].

[38] R. C. Gonzalez, R. E. Woods, "Digital image processing," Third edition, Prentice Hall, ISBN 9780131687288, pp. 402-407, 528-538, 2006.

[39] The Netfilter.org Project, Available online: http://www.Netfilter.org/, [Last Accessed: 1st March 2009].

[40] Wireshark, Available online: http://www.wireshark.org/ , [Last Accessed on 21st Jan. ' 2009].

[41] M. Hammami, Y. Chahir and L. Chen, "WebGuard: Web Based Adult Content Detection and Filtering System," *IEEE/WIC International Conference on Web Intelligence,* Vol. 2, pp. 574-578, 13-17, Oct. 2003.

[42] D. Forsyth and M. Fleck, "Automatic Detection of Human Nudes," *International Journal of Computer Vision*, pp.63-77, Aug. 1999.

[43] M. Adams and F. Kossentini, "A Software-based JPEG-2000 Codec Implementation," *International Conference on image processing*, Vol.2, pp.53-56, 2000.

[44] International Telecommunications Union, "Information technology – Digital compression and coding of continues-tone still images – Requirements and guidelines," Available online: http://www.itu.int/net/home/index.aspx, [Last Accessed: 3rd Jan. 2009].

[45] S. Stuijk, "Design and Implementation of JPEG Decoder", *Practical Training Report,* Eindhoven University of Technology, Leuven – Belgium, 2001.

[46] B. J. Oommen, and L. Rueda, "Stochastic Learning-based Weak estimation of Multinomial Random Variables and its Applications to Pattern Recognition in Non-stationary Environments," *Pattern Recognition*, Vol. 39, pp. 328-341, 2006.

[47] B. J. Oommen and L. Rueda L, "On Utilizing Stochastic Learning Weak Estimators for Training and Classification of Patterns with Non-stationary Distributions," *Proc. of the 28th German Conference on Artificial Intelligence*, Koblenz, Germany, Springer, pp. 107-120, 2005.

APPENDIXES

Appendix A:

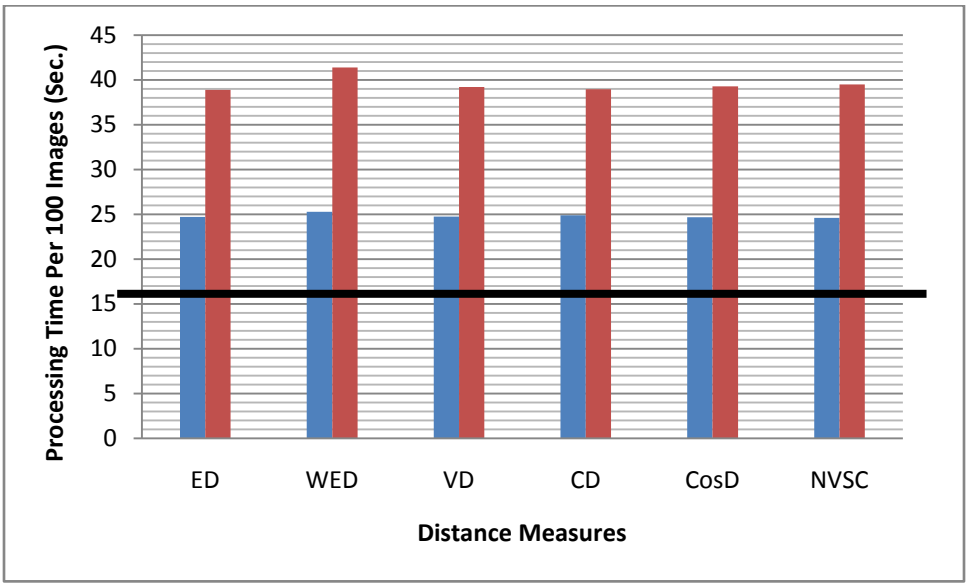
Appendix A-1: Standard JPEG image markers. (Table courtesy of http://www.impulseadventure.com/)

Hex	Marker	Marker Name	Description
0x FFC0	SOF0	Start of Frame 0	Baseline DCT
0x FFC1	SOF1	Start of Frame 1	Extended Sequential DCT
0x FFC2	SOF2	Start of Frame 2	Progressive DCT
0x FFC3	SOF3	Start of Frame 3	Lossless (sequential)
0x FFC4	DHT	Define Huffman Table	N/A
0x FFC5	SOF5	Start of Frame 5	Differential sequential DCT
0x FFC6	SOF6	Start of Frame 6	Differential progressive DCT
0x FFC7	SOF7	Start of Frame 7	Differential lossless (sequential)
0x FFC8	JPG	JPEG Extensions	N/A
0x FFC9	SOF9	Start of Frame 9	Extended sequential DCT, Arithmetic coding
0x FFCA	SOF10	Start of Frame 10	Progressive DCT, Arithmetic coding
0x FFCB	SOF11	Start of Frame 11	Lossless (sequential), Arithmetic coding
0x FFCC	DAC	Define Arithmetic Coding	
0x FFCD	SOF13	Start of Frame 13	Differential sequential DCT, Arithmetic coding
0x FFCE	SOF14	Start of Frame 14	Differential progressive DCT, Arithmetic coding
0x FFCF	SOF15	Start of Frame 15	Differential lossless (sequential), Arithmetic coding

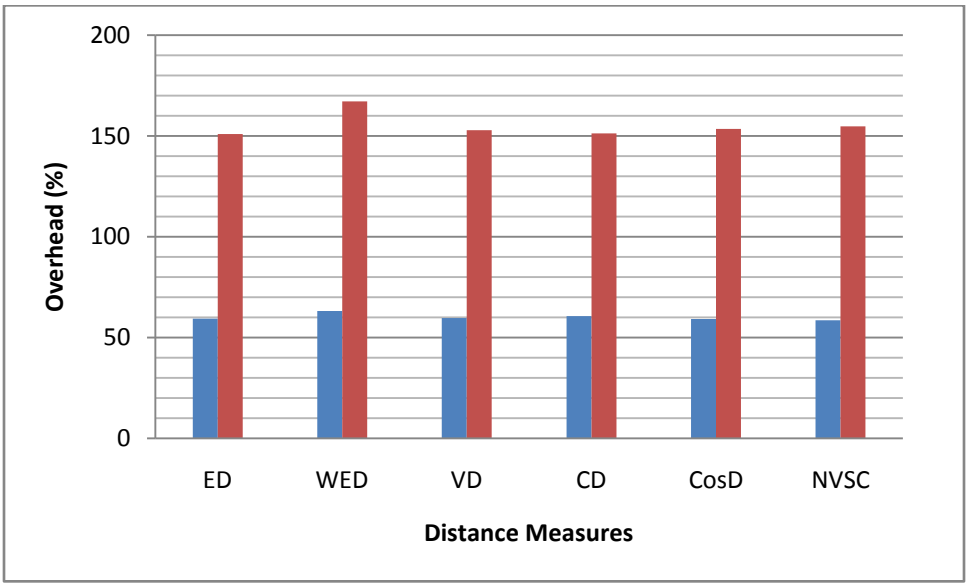
0x FFD0	RST0	Restart Marker 0	N/A
0x FFD1	RST1	Restart Marker 1	N/A
0x FFD2	RST2	Restart Marker 2	N/A
0x FFD3	RST3	Restart Marker 3	N/A
0x FFD4	RST4	Restart Marker 4	N/A
0x FFD5	RST5	Restart Marker 5	N/A
0x FFD6	RST6	Restart Marker 6	N/A
0x FFD7	RST7	Restart Marker 7	N/A
0x FFD8	SOI	Start of Image	N/A
0x FFD9	EOI	End of Image	N/A
0x FFDA	SOS	Start of Scan	N/A
0x FFDB	DQT	Define Quantization Table	N/A
0x FFDD	DRI	Define Restart Interval	N/A
0x FFE0	APP0	Application Segment 0	JFIF - JFIF JPEG image AVI1 - Motion JPEG (MJPG)
0x FFE1	APP1	Application Segment 1	EXIF Metadata, TIFF IFD format, JPEG Thumbnail (160x120) Adobe XMP
0x FFF7	JPG7 SOF48	JPEG Extension 7 JPEG-LS	Lossless JPEG
0x FFF8	JPG8 LSE	JPEG Extension 8 JPEG-LS Extension	Lossless JPEG Extension Parameters
0x FFFE	COM	Comment	N/A
0x FFFF	Stuff	Stuff Byte	Representation of 0xFF in data stream

Appendix B: Processing times and overhead.

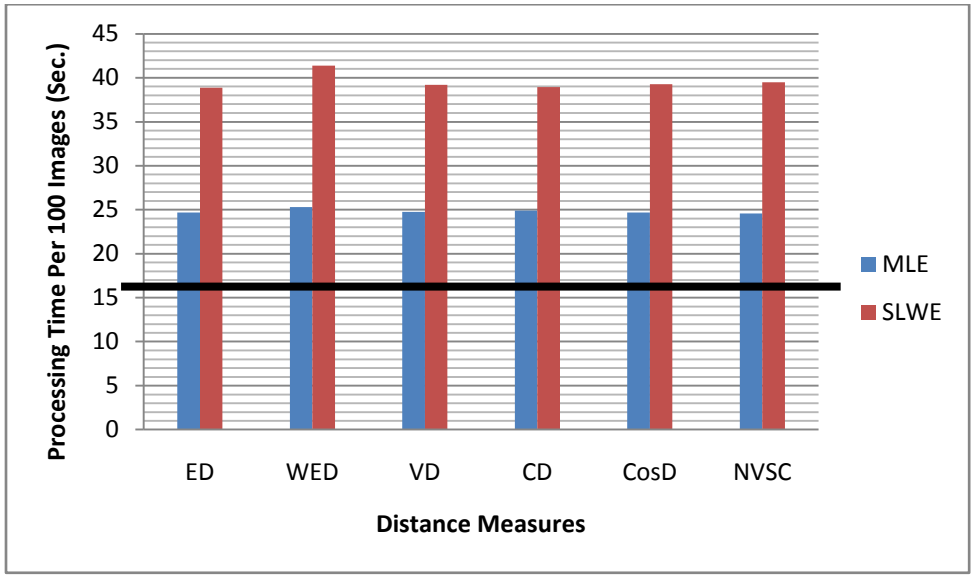
Appendix B- 1: Processing time when considering all colour components of the RGB colour space per 100 images, black line shows processing time without filter enabled.



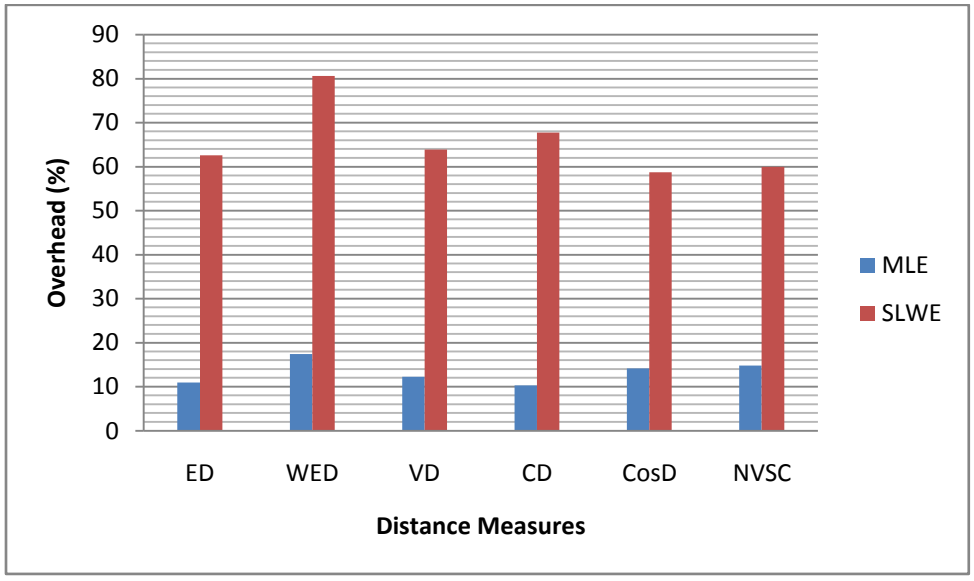
Appendix B- 2: Percent overhead when considering all colour components of the RGB colour space per 100 images.



Appendix B- 3: Processing times of 100 images when only the B colour component; black line shows processing time without filter enabled.

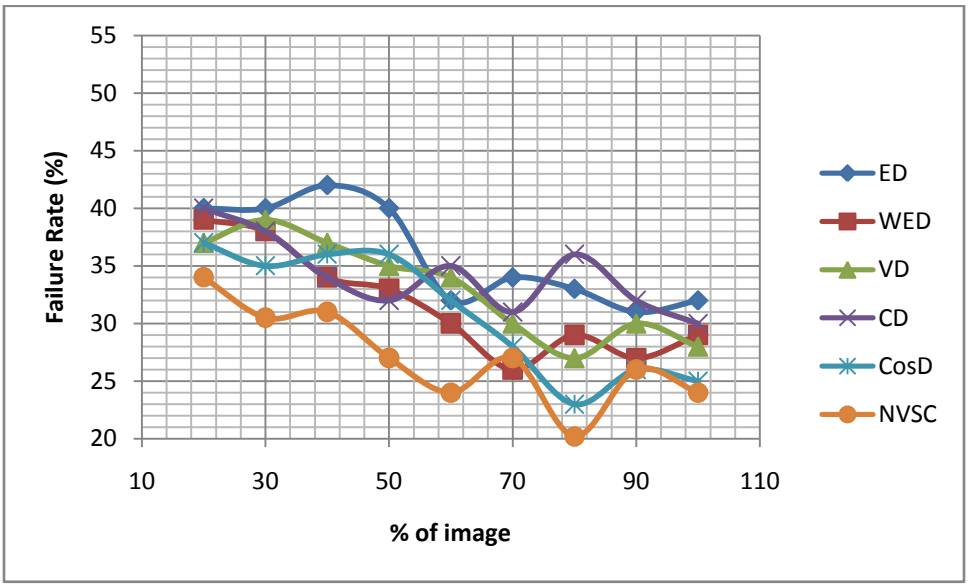


Appendix B- 4: Percent overhead when considering only the B colour component

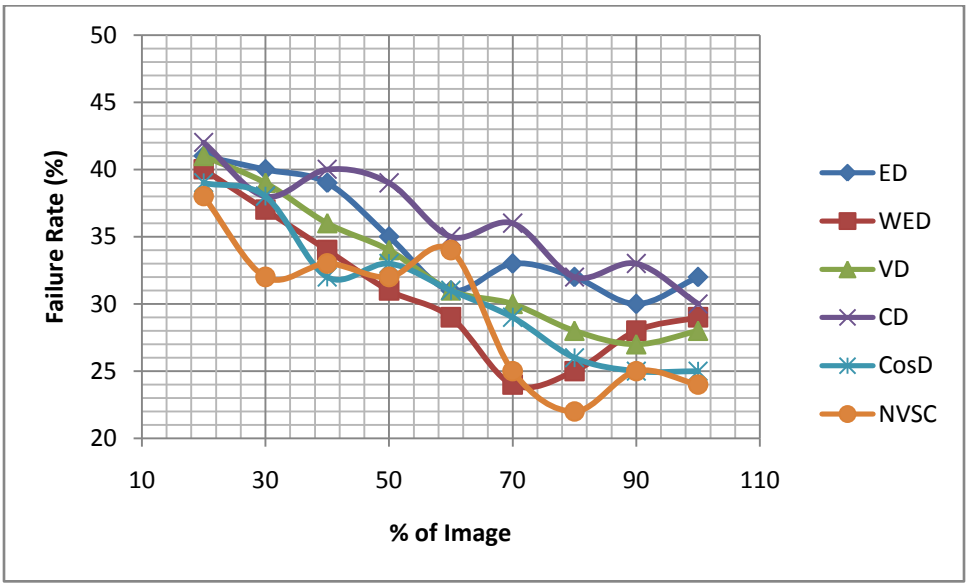


Appendix C: Success and error rates

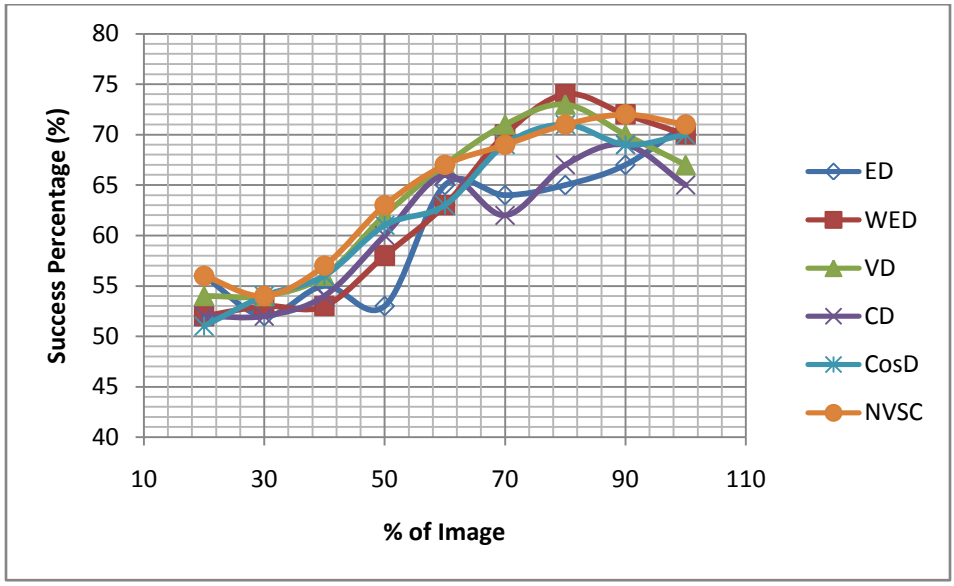
Appendix C- 1: Error rate using height scan; all test images are non-nude (SLWE using RGB colour space)



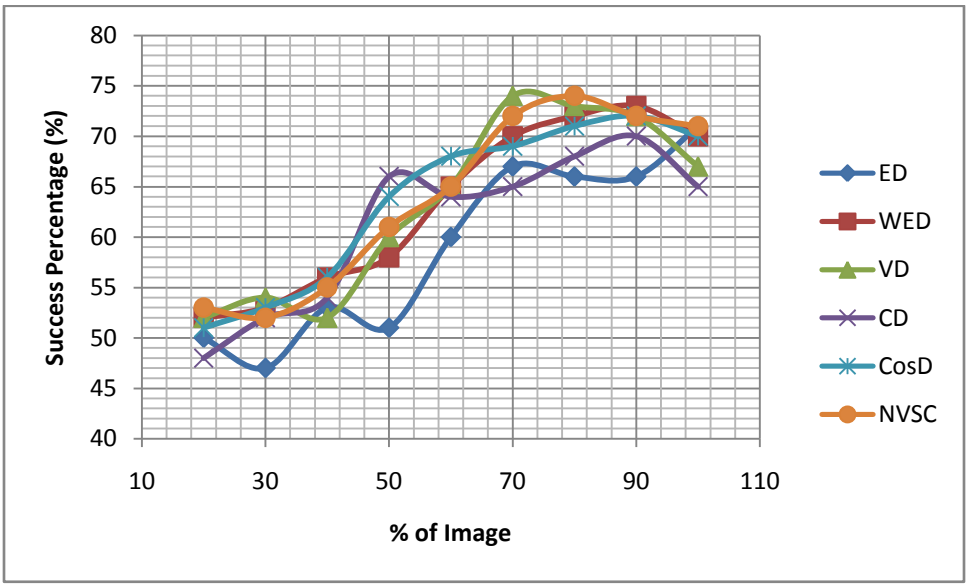
Appendix C- 2: Error rate using radius scan; all test images are non-nude (SLWE using RGB colour space).



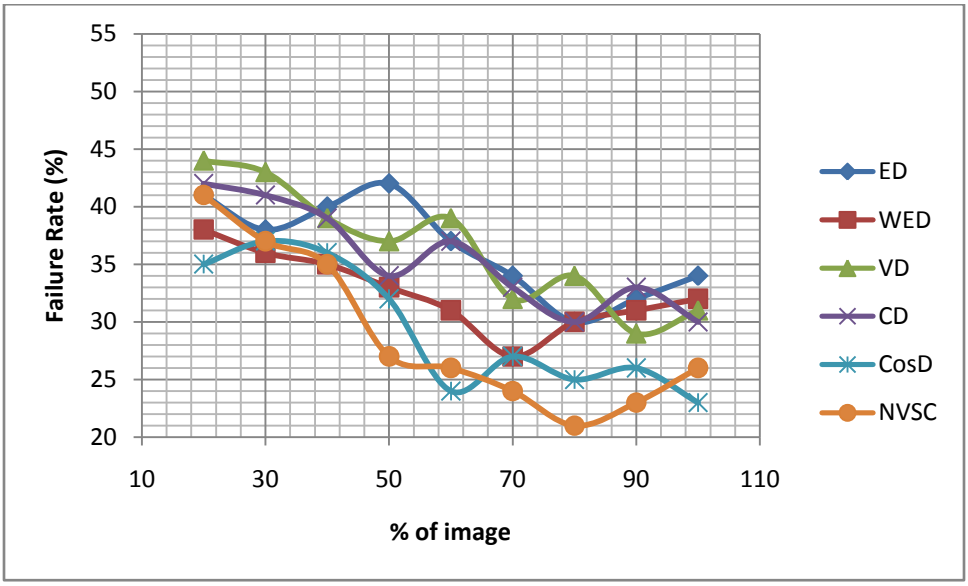
Appendix C- 3: Success rate using height scan; all test images are nude (MLE using only the B component of RGB colour space).



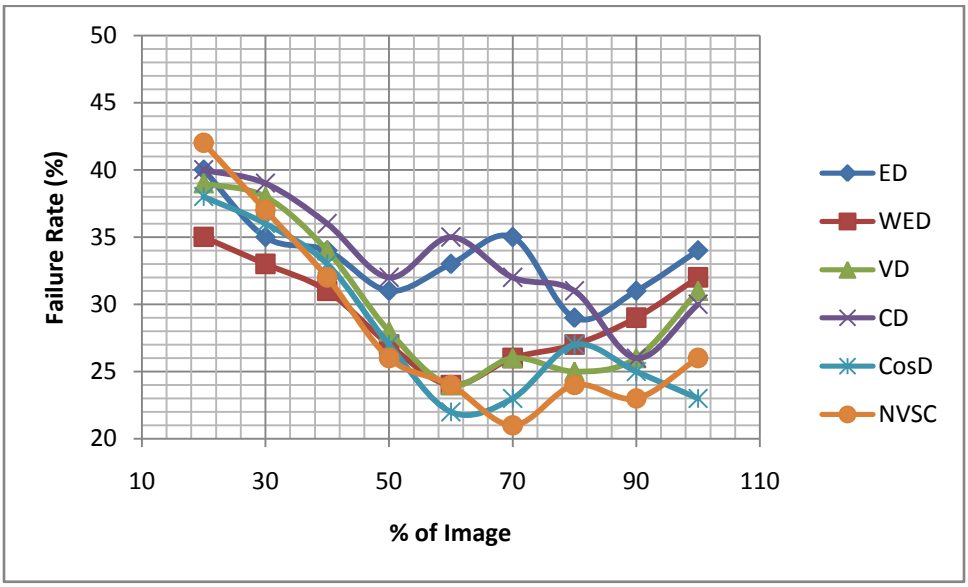
Appendix C- 4: Success rate using radius scan; all test images are nude (MLE using only the B component of RGB colour space).



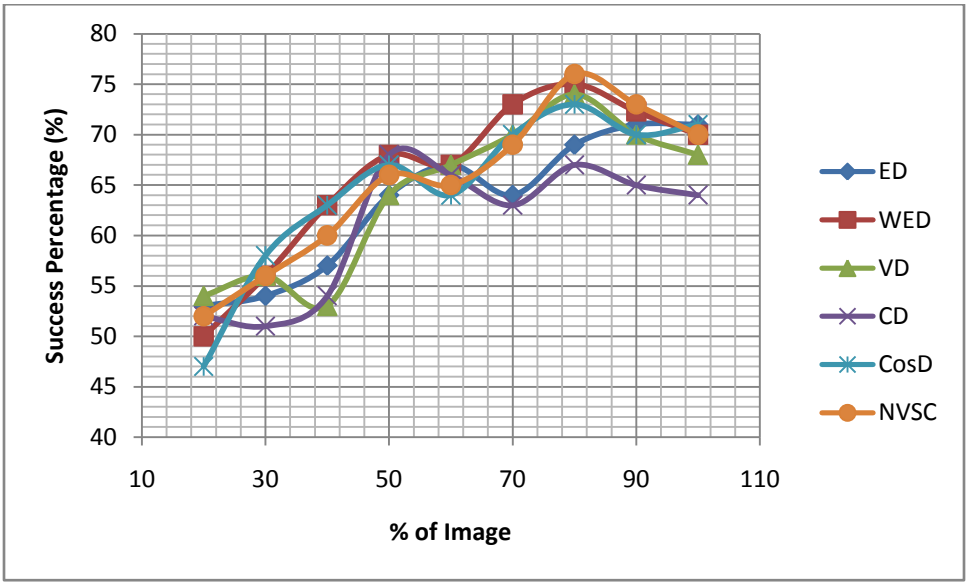
Appendix C- 5: Error rate using height scan; all test images are non-nude (MLE using only the B component of RGB colour space).



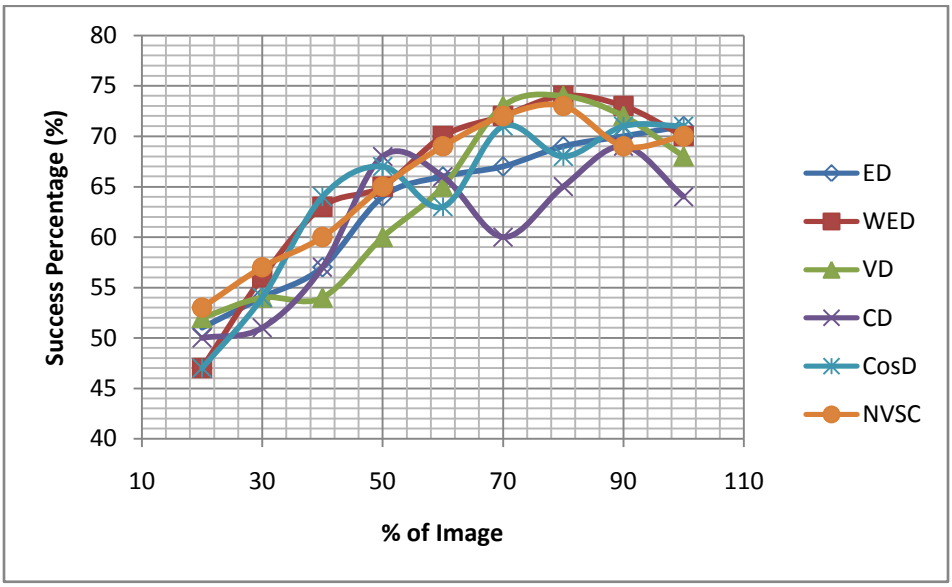
Appendix C- 6: Error rate using radius scan; all test images are non-nude (MLE using only the B component of RGB colour space).



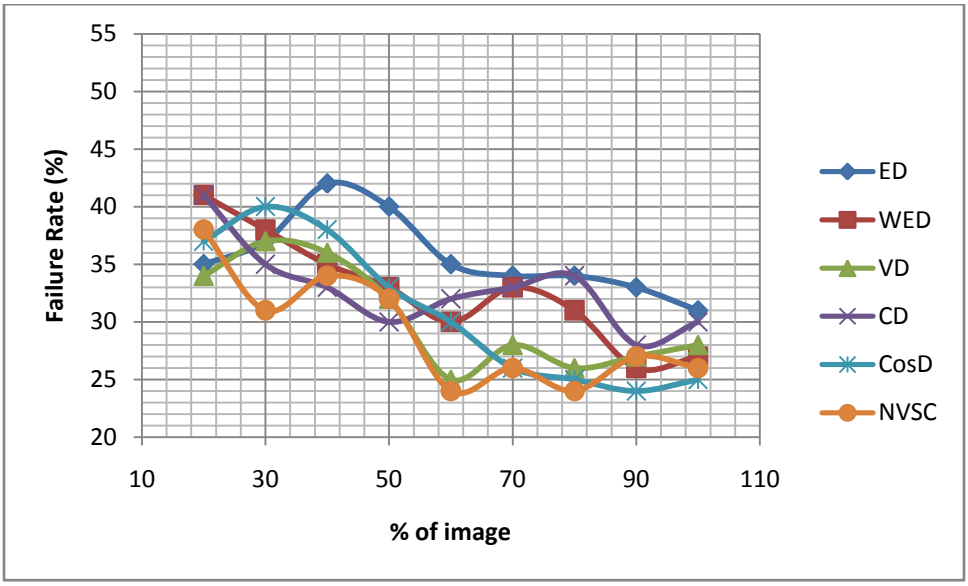
Appendix C- 7: Success rate using height scan; all test images are nude (SLWE using only the B component of RGB colour space).



Appendix C- 8: Success rate using radius scan; all test images are nude (SLWE using only the B component of RGB colour space).



Appendix C- 9: Error rate using height scan; all test images are non-nude (SLWE using only the B component of RGB colour space).



Appendix C- 10: Error rate using radius scan; all test images are non-nude (SLWE using only the B component of RGB colour space).