Computer Science & Information Technology 45

David C. Wyld
Jan Zizka (Eds)

Computer Science & Information Technology

Second International Conference on Computer Science & Engineering
(CSEN 2015)
Dubai, UAE, August 28~29, 2015

AIRCC

Volume Editors

David C. Wyld,
Southeastern Louisiana University, USA
E-mail: David.Wyld@selu.edu

Jan Zizka,
Mendel University in Brno, Czech Republic
E-mail: zizka.jan@gmail.com

ISSN: 2231 - 5403
ISBN: 978-1-921987-42-7
DOI : 10.5121/csit.2015.51401 - 10.5121/csit.2015.51410

Typesetting: Camera-ready by author, data conversion by NnN Net Solutions Private Ltd., Chennai, India

Preface

The Second International Conference on Computer Science & Engineering (CSEN 2015) was held in Dubai, UAE, during August 28~29, 2015. The International Conference on Artificial Intelligence and Soft Computing (AISO 2015), The International Conference of Networks, Communications, Wireless and Mobile Computing (NCWC 2015) and The International Conference on Signal and Pattern Recognition (SIPR 2015) were collocated with the CSEN 2015. The conferences attracted many local and international delegates, presenting a balanced mixture of intellect from the East and from the West.

The goal of this conference series is to bring together researchers and practitioners from academia and industry to focus on understanding computer science and information technology and to establish new collaborations in these areas. Authors are invited to contribute to the conference by submitting articles that illustrate research results, projects, survey work and industrial experiences describing significant advances in all areas of computer science and information technology.

The CSEN-2015, AISO-2015, NCWC-2015, SIPR-2015 Committees rigorously invited submissions for many months from researchers, scientists, engineers, students and practitioners related to the relevant themes and tracks of the workshop. This effort guaranteed submissions from an unparalleled number of internationally recognized top-level researchers. All the submissions underwent a strenuous peer review process which comprised expert reviewers. These reviewers were selected from a talented pool of Technical Committee members and external reviewers on the basis of their expertise. The papers were then reviewed based on their contributions, technical content, originality and clarity. The entire process, which includes the submission, review and acceptance processes, was done electronically. All these efforts undertaken by the Organizing and Technical Committees led to an exciting, rich and a high quality technical conference program, which featured high-impact presentations for all attendees to enjoy, appreciate and expand their expertise in the latest developments in computer network and communications research.

In closing, CSEN-2015, AISO-2015, NCWC-2015, SIPR-2015 brought together researchers, scientists, engineers, students and practitioners to exchange and share their experiences, new ideas and research results in all aspects of the main workshop themes and tracks, and to discuss the practical challenges encountered and the solutions adopted. The book is organized as a collection of papers from the CSEN-2015, AISO-2015, NCWC-2015, SIPR-2015.

We would like to thank the General and Program Chairs, organization staff, the members of the Technical Program Committees and external reviewers for their excellent and tireless work. We sincerely wish that all attendees benefited scientifically from the conference and wish them every success in their research. It is the humble wish of the conference organizers that the professional dialogue among the researchers, scientists, engineers, students and educators continues beyond the event and that the friendships and collaborations forged will linger and prosper for many years to come.

David C. Wyld
Jan Zizka

Organization

General Chair

Natarajan Meghanathan	Jackson State University, USA
Dhinaharan Nagamalai	Wireilla Net Solutions PTY LTD, Australia

Program Committee Members

Abdelkrim Khireddine	University of Bejaia, Algeria
Abdul Kadhir Ozcan	KTO karatay University, Turkey
Abe Zeid	Northeastern University, USA
Ahmed Taisser	Cairo University, Egypt
Ali Dorri	Islamic Azad University, Iran
Ali Hussein Mohammed	Alexandria University, Egypt
Alireza Afshari	Islamic Azad University, Iran
Amani Samha	Amani samha researcher QUT, Australia
Ankit Chaudhary	Truman State University, USA
Asadollah Shahbahrami	University of Guilan, Iran
Ashraf A. Shahin	Cairo Univeristy, Egypt
Atheer Yousif Oudah	Thi-Qar University, Iraq
Bai Li	Woodside Energy Ltd, Australia
Barbaros Preveze	Cankaya University, Turkey
Braham Barkat	The petroleum Institute, Saudi Arabia
Chandan Kumar Karmakar	University of Melbourne, Australia
Cheng fang	Zhejiang University, China
Chih-Lin Hu	National Central University, Taiwan
Chin-Chih Chang	Chung Hua University,Taiwan
Chiranjib Sur	University of Florida, US
Christian Esposito	National Research Council, Italy
Dac-Nhuong Le	Haiphong University, Vietnam
Danda B.Rawat	Georgia Southern University, USA
Daniel Dacuma Dasig	Jose Rizal University, Philippines
Derya Birant	Dokuz Eylul University, Turkey
Djouani	University Paris-Est Créteil, France
Ehsan Heidari	Islamic Azad University, Iran
Emilio Jimenez Macias	University of La Rioja, Spain
Fan Zhang	University of South Carolina, Columbia
Farhan	University of Indonesia, Indonesia
Fatemeh Deregeh	Shahid Bahonar University of Kerman, Iran
Fatih Korkmaz	cankiri karatekin university, Turkey
Francisco Macia Perez	University of Alicante, Spain
Gullanar M Hadi	Salahaddin University, Hawler, Iraq
Hacene Belhadef	University of Constantine 2, Algeria
Hamdi M	National Engineering School of Tunis, Tunisia
Hao-En Chueh	Yuanpei University, Taiwan, Republic of China

Technically Sponsored by

Networks & Communications Community (NCC)

Computer Science & Information Technology Community (CSITC)

Digital Signal & Image Processing Community (DSIPC)

Organized By

Academy & Industry Research Collaboration Center (AIRCC)

TABLE OF CONTENTS

Second International Conference on Computer Science & Engineering (CSEN 2015)

International Conference on Artificial Intelligence and Soft Computing (AISO 2015)

International Conference of Networks, Communications, Wireless and Mobile Computing (NCWC 2015)

International Conference on Signal and Pattern Recognition (SIPR 2015)

FRACTAL ANALYSIS OF GOOD PROGRAMMING STYLE

Ron Coleman and Pritesh Gandhi

Computer Science Department, Marist College, Poughkeepsie, New York, USA
roncoleman@marist.edu pritesh.gandhi1@marist.edu

ABSTRACT

This paper studies a new, quantitative approach using fractal geometry to analyse basic tenets of good programming style. Experiments on C source of the GNU/Linux Core Utilities, a collection of 114 programs or approximately 70,000 lines of code, show systematic changes in style are correlated with statistically significant changes in fractal dimension (P≤0.0009). The data further show positive but weak correlation between lines of code and fractal dimension (r=0.0878). These results suggest the fractal dimension is a reliable metric of changes that affect good style, the knowledge of which may be useful for maintaining a code base.

KEYWORDS

Good programming style, fractal dimension, GNU/Linux Core Utilities

1. INTRODUCTION

Good programming style is a way of writing source code. Although different style guides have different conventions, a survey of contemporary texts [1] [2] [3] [4] finds general agreement on three basic rules: use proper indentation, include documentation, and choose meaningful or mnemonic names. While style guides stress the importance of good style, especially for maintenance purposes, "good" is a value word and "style" connotes, among other things, form and taste. In other words, we propose source has potential elegance as a work of art like a painting or photograph and indeed, any given programming style, including an indecorous one, may be readily accessible without an in-depth understanding of how the code works or even what it does. In this view, source has aesthetic or *sensori-emotional* qualities.

We are not suggesting aesthetic appeal in code should be an overarching goal of software, only that it plays a role in crafting and maintaining code as a best practice. Yet aesthetics present challenges. According to a modernist, Kantian view [5], aesthetics in general and notions of beauty and matters of taste in particular are thought to be subjective, relative, and presumably beyond the pale of automation. However, software engineers have sidestepped these dilemmas, asking not what *is* beauty in source but rather what is *knowable* about such beauty (e.g., good programming style), which can be incorporated in programs like the GNU/Linux command, *indent* [6], which beautifies C source by refactoring indentation, comments, and spacing. Tools like *indent* are a staple of modern software engineering. Unfortunately, these tools do not quantify the value of their beautifying regimes, as a consequence developers have had to resort to

David C. Wyld et al. (Eds) : CSEN, AISO, NCWC, SIPR - 2015
pp. 01–10, 2015. © CS & IT-CSCP 2015 DOI : 10.5121/csit.2015.51401

anecdotal arguments rather than metrics to reason about aesthetic outcomes, and no research effort heretofore has investigated the problem or its opportunities.

In this paper, we study a new, quantitative way to analyse basic tenets good programming style using fractal geometry [7]. Fractals are often associated with beauty in nature and human designs [8]. Furthermore since fractals are self-similar and scale-invariant, we hypothesized a fractal approach might be inherently robust for handling distributions of source sizes.

Experiments with the C source code of the GNU/Linux Core Utilities [9], 114 commands of the Linux shell or about 70,000 lines of code (LOC), show systematic changes in programming style are correlated with statistically significant changes ($P \leq 0.0002$) in fractal dimension [10]. The data further show that while the baseline sizes of C source files vary widely, there is a positive but weak correlation with fractal dimension (r=0.0878). These data suggest the fractal dimension is a reliable metric of changes in source that affect good style, the knowledge of which may be useful for maintaining a code base.

2. RELATED WORK

Aesthetic value in source is not the same as readability [11] [12], although the two are related. The latter is more about comprehending code whereas the former, appreciating it, *l'art pour l'art*. Beauty in source is also not the same as functional complexity [13]. Complexity relates to design and efficiency in algorithms and data structures, which may have appeal in a conceptual, though not necessarily a visual sense, although here again there is overlap. *Beautiful Code* [14] explores just this sort of conceptual aesthetic, not only in source but also in debugging and testing which are not subjects we consider. Gabriel [15] argues against clarity and conceptual beauty as primary goals of software in favour of what the author calls "habitability." Yet comfort with the code is independent of style since programmers might forgo style best practices as long as they can live with it, whereas our starting point *is* good style. The fractal dimension has been applied to a wide range of disciplines, though not software development [16]. Our code depends on Fractop [17], a Java library originally developed to categorize neural tissue. We have reused this library to analyze source code. Some researchers have employed the fractal dimension to study paintings of artists [18]; others working in a similar vein have used the fractal dimension to authenticate Jackson Pollack's "action paintings" [19] [20]. Still others have used the fractal dimension to examine aesthetic appeal in artificially intelligent path finding in videogames [21] [22] [23]. An investigation of Scala repositories on GitHub.com found sources are organized according to power-law distributions [24] [25] but that effort did not consider style. Kokol, et al, [26] [27] [28] reported evidence of fractal structure and long-range correlations in source; however, they were investigating not style but fine details, character, operator, and string patterns in a small sample of randomly generated Pascal programs. We study style in a moderate size sample of highly functional C programs.

3. METHODS

We use a multi-phase operation to process a single source file: 1) beautify or de-beautify the source style, if necessary; convert the result to an in-memory representation called an *artefact*; 3) calculate the fractal dimension of the artefact.

To beautify the source in phase 1, we use a combination of the GNU/Linux *indent* command and a kit we developed called *Mango* [29] (see below). The *indent* manual page [6] gives input options for beautifying the source according to four distinct C styles: GNU, K&R (Kernighan and Ritchie), Berkeley, and Linux (kernel). They affect indentation, spacing, and comments and differences can be found in the manual page. The command, *indent*, does not, however, change mnemonics.

Mango is a kit written in Scala, C, and to drive the experiments, Korn shell scripts. During the first phase of processing, Mango mostly does the reverse of *indent*: it "mangles" or de-beautifies C source and outputs new source as we discuss below.

3.1. Base lining measurements

To get baseline measurements of the source, Mango skips phase 1 and sends the unmodified source directly to phases 2 and 3 to generate the artefact and calculate the fractal dimension, respectively.

3.2. De-beautifying source

When de-beautifying source in phase 1, Mango does one of the following: remove indentation, randomize indentation, remove comments, or make the names of variables, functions, macros, and labels less mnemonic. To remove indentation, Mango trims each line of spaces. To randomize the indentation, Mango inserts a random number of spaces to the beginning of the line. To remove comments, Mango strips the file of both block (/* … */) and line (//) comments. Finally, to make names less mnemonic, Mango shortens them according to the algorithm below.

3.3. Non-mnemonic algorithm

The algorithm to shorten names requires two passes over the source. During the first pass Mango filters key words, compiler directives, library references, names with less than a minimum length (l=3), and names appearing less than a minimum frequency (n=3). For names that get through these filters, Mango calculates new, non-mnemonic names as follows. If a name has at least one under bar ("_"), Mango splits the name along the under bar and recombines the first letter of each subsequent sub-name with the whole first sub-name followed by an under bar. If a name is uppercase name, Mango uses every other letter to reform the name, effectively, cutting the name in half. If a name is neither of these, it shortens the name by half. Mango puts the old name and the new name in a database for lookup and substitution back into the source during the second pass. The table below gives some examples of how the algorithm works.

Table 1. Example changes by non-mnemonic algorithm

Old name	New name
i	i
T_FATE_INIT	T_FI
NOUPDATE	NUDT
linkname	link

3.4. Mnemonic algorithm

Mango also has a beautify mode of phase 1 to make names more mnemonic. Mango does not, of course, know the intention of programmers or semantics of names. However, it can simulate these by lengthening names. The algorithm to lengthen names is similar to the one to shorten them. During the first pass Mango collects appropriately filtered candidate names of a maximum length (*l*=3) and with a minimum frequency (*n*=3). Mango makes these names a maximum of length of four by repeating the letters in the name or adding an under bar after the name. The table below gives some examples of how the algorithm works.

Table 2. Example changes by the mnemonic algorithm

Old name	New name
loop	loop
foo	foo_
go	gogo
i	iiii

3.5. Artefact generation

Phase 2 of Mango converts an input source file it to an artefact, which has one of two types of encodings: *literal* and *block.*

With literal encoding, the flat text of the source is written to a buffered image using a graphics context. The text is Courier New, ten-point, plain style, and black foreground over a white background with ten-point line height. In this case, the artefact looks identical the flat text except it's in bitmap form.

With block encoding, each character in the input is written to the graphics context as "blocks" or 8×10 (pixels) black filled rectangles over a white background with two pixels between each rectangle. Spaces are 10×10 pixels. A block artifact resembles the source but in digital outline.
Block encoding has two advantages. It makes the artefact more robust, more independent language. Similarly, it makes the mnemonic and non-mnemonic algorithms more robust. In fact, for these algorithms with block encoding, only the length of the name is relevant, not the name itself.

The figure below is an example of a simple C program.

```
#include <stdio.h>
int main(int argc, char** argv) {
  printf("Hello, world!");
  return 0;
}
```

Figure 1. Simple C file which is identical to its literal artefact encoding except in bitmap form

A literal artifact looks identical to the figure above except it is a bitmap.

The figure below shows the same C program as a block artifact.

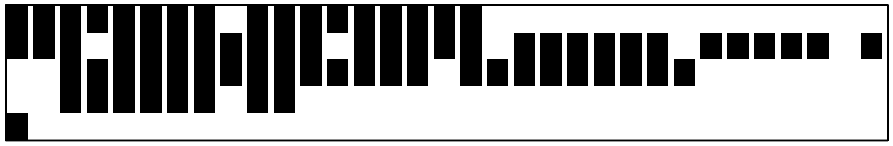


Figure 2. Same C file as an artefact with block encoding

As the reader can see from the figure above, all the language details have been “blocked”. Only the digital outline persists.

3.6. Fractal dimension calculation

The third and final phase of Mango measures the fractal dimension of the artefact. Mandelbrot [9] described fractals as geometric objects, which are no-where differentiable, that is, textured, and self-similar at different scales. We use the geometric interpretation based on reticular cell counting or the box counting dimension. We choose this method for two reasons. Firstly, the box counting dimension is conceptually and computationally straightforward. Secondly, Fractop [x] provides a tested, high quality implementation.

Mandelbrot also said fractal objects have fractional dimension, *D*, namely, a non-whole number called the fractional dimension. Mathematically, *D* is given by the Hausdorff dimension [15]:

$$D(S) = \lim_{\varepsilon \to \infty} \frac{log N_{\varepsilon}}{\log\left(\frac{1}{\varepsilon}\right)} \tag{1}$$

where *S* represents a set of points on a surface (e.g., coastlines, brush strokes, source lines of code, etc.), ε is the size of the measuring tool or ruler and $N_{\varepsilon}(S)$ is the number of self-similar objects or subcomponents covered by the measuring tool. For fractal objects, log $N_{\varepsilon}(S)$ will be greater than log (1/ε) by a fractional amount. If the tool is a uniform grid of square cells, then a straight line passes through twice as many cells if the cell length is reduced by a factor of two. A fractal object passes through more than twice as many cells.

The artefact is *S* from Equation 1. Mango uses the Fractop default grid sizes of 2, 3, 4, 6, 8, 12, 16, 32, 64, and 128 measured in pixels for ε. For any given input artefact, Mango returns *D*, which is the slope of the line of the log proportion of cells intersected by the surface increases as log cell size decreases.

4. EXPERIMENT DESIGN

The GNU/Linux Core Utilities version 8.10 [8] comprise 114 dot C source files. First, we generated descriptive statistics for this test bed for number of files and LOC.

We then ran three experiments as follows

1. Established baseline *D* using the original, unmodified C files with literal and block artefact encodings.

2. Treat the source with de-beautifying regimes using Mango to i) remove indentation, ii) randomize indentation by 0-20 spaces, iii) randomize indentation by 0-40 spaces, iv) make names non-mnemonic, and v) remove comments.

3. Treat the source with beautifying regimes using Mango to i) make names more and using GNU/Linux *indent* to refactor the source with ii) GNU, iii) K&R, iv) Berkeley, and v) Linux style settings.

We observed the frequency and direction in which *D* changes relative to the baseline. We computed the percentage change and the one-tailed *P*-value using the Binomial test [30]. We also measured the rank correlation coefficient, Spearman's rho [30], between the baseline *D* and lines of code over all source files.

5. RESULTS

The table below gives the test bed summary statistics. The range of LOC is fairly wide, from files with just two lines to several thousand lines.

Table 3. Test bed summary statistics

Files	114
Total LOC	69,722
Median LOC	356
Maximum LOC	4,733
Minimum LOC	2

The table below gives the baseline fractal dimension values for literal and block encodings.

Table 4. Baseline analysis

	Literal	Block
Median *D*	1.4592	1.6500
Maximum *D*	1.5448	1.7176
Minimum *D*	0.9836	1.4011
r (LOC v. *D*)	0.0878	0.0878

5.1 De-beautifying treatments

The tables below give the direction and the frequency of changes *D* decreases in relation to the baseline. As the reader can see the fractal dimension decreases in each case with a small difference between literal and block encoded artefacts. Removing indents is statistically significant, however, as a contrarian indicator. In other words, rather than decreasing *D*, it increases it in relation to the baseline. We explore this matter further below.

Table 5. Changes in *D* in relation to the baseline with literal encoding

Treatment	Dir.	Freq.	Rate	P
Random indents 0-20	down	112	98%	<0.0001
Random indents 0-40	down	109	96%	<0.0001
Remove indents	up	107	94%	<0.0001
Remove comments	down	82	72%	<0.0001
Non-mnemonic	down	104	91%	<0.0001

Table 6. Changes in *D* in relation to the baseline with block encoding

Treatment	Dir.	Freq.	Rate	P
Random indents 0-20	down	113	99%	<0.0001
Random indents 0-40	down	113	99%	<0.0001
Remove indents	up	107	94%	<0.0001
Remove comments	down	112	98%	<0.0001
Non-mnemonic	down	106	93%	<0.0001

5.2 Beautifying treatments

The tables below give the direction and the frequency of changes *D* decreases in relation to the baseline.

Table 7. Changes in *D* in relation to the baseline with literal encoding

Treatment	Dir.	Freq.	Rate	P
GNU style	up	100	88%	<0.0001
K&R style	up	105	92%	<0.0001
Berkeley style	up	74	65%	0.0009
Linux style	up	106	93%	<0.0001
Mnemonic	up	97	85%	<0.0001

Table 8. Changes in *D* in relation to the baseline with block encoding

Treatment	Dir.	Freq.	Rate	P
GNU style	up	112	98%	<0.0001
K&R style	up	104	91%	<0.0001
Berkeley style	up	78	68%	<0.0001
Linux style	up	105	92%	<0.0001
Mnemonic	up	99	87%	<0.0001

5.3 No indentation as contrarian indicator

The experimental results in section 5.1, “De-beautifying treatments,” removed indentation on all the source lines and we found *D* increased. We hypothesized that if removing indentation were a contrary indicator, we expect *D* to rise from the baseline (0% rate) to complete indentation removal (100% rate). The null hypothesis is no change in *D* is affected by the removal rate. To test the null hypothesis, namely, no change in *D* with change in removal rate, we examined several files and found we could reject the null, at least on a subset of typical size files. For instance, mktemp.c has 358 LOC, which is very close to the median size file. We removed the indentation on randomly selected lines at 75%, 50%, and 25% rates and measured *D* in ten trials using literal encoding. The data for mktemp.c is in the table below is typical for other programs we examined.

Table 9 *D* for different random remove rates over ten trials for mktemp.c

	Indentation removal rate		
Trial	25%	50%	75%
1	1.468205428	1.470438295	1.476648907
2	1.46463698	1.472219091	1.47721244
3	1.465692458	1.470056954	1.475848552
4	1.465102815	1.47256331	1.479550183
5	1.464691894	1.469024252	1.477846232
6	1.464413407	1.470376845	1.480434004
7	1.465313286	1.474732486	1.481568639
8	1.466252928	1.470800863	1.480060737
9	1.469609632	1.470203698	1.474179211
10	1.467231153	1.468487205	1.480865379
Median	1.465502872	1.47040757	1.478698207

The chart below shows the plot with the median values for 25%, 50%, and 75% removal rates, the baseline (0%), and complete removal (100%).

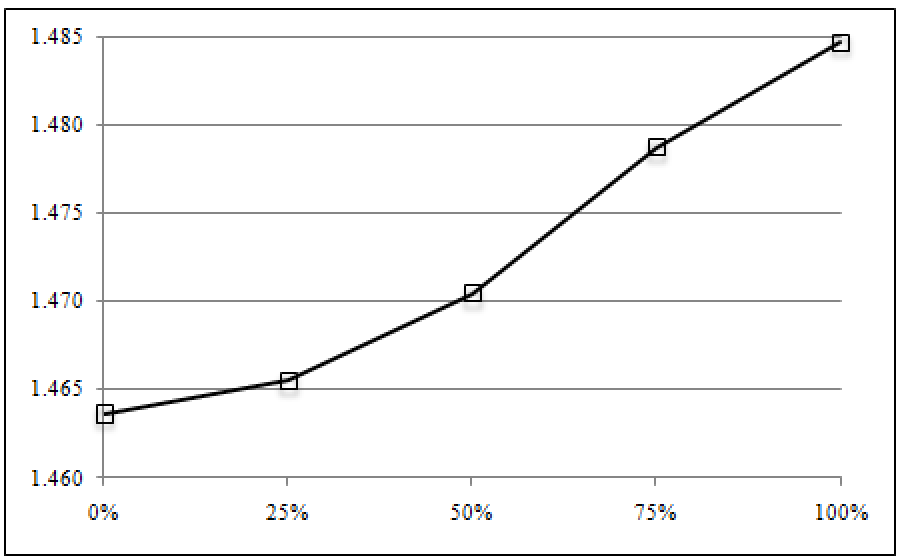


Figure 3 The rate of indentation removal rate vs. *D* for mktemp.c where 0% is the baseline and 100% is removal of all indentation.

6. DISCUSSION

The first observation we make is generally $D^{\text{literal}} < D^{\text{block}}$. This makes sense since the block encoding covers more surface area, *S*, in the artefact than the literal encoding. Our preference is for block encoding because of its robustness we mentioned earlier. Nevertheless the pattern of results is consistent between literal and block encoding. When we de-beautify the source, *D* decreases; when we beautify the source, *D* increases.

The exception, we noted, is the removal of all indentation. Yet Figure 1 suggests that removing indentation is a contrarian indicator of style. We believe the contrariness is a peculiar property of the fractal dimension. That is, keeping in mind that D=2 means there is no texture and we have a completely covered surface of a solid colour, the larger *D* for removing indentation implies greater surface area. Thus, having all the text aligned on the left gives a more compact, and thus complete, surface.

All the beautifying treatments increase in *D*. The indent command programmed with Linux style is the most effective for raising *D* and Berkeley style, the least effective.

What is most interesting is that since the GNU/Linux Core Utilities were presumably written with the GNU style guide, the GNU style-beautifying regime nonetheless increases *D*. If changes in *D* are represent changes in style as the data suggests, then it appears there may be room yet for style improvements in the Core Utilities.

This observation offers insight into how to formulate a relative aesthetic value. Consider, for instance, the conflict between regimes that beautify code and increase *D* and the contrarian effect of removing all indentation, which de-beautify the code but also increase *D*. One way to resolve this is to randomly sample the removal of indentation at different rates, measure *D* for each rate as we did above, and test the slope of the line. If it is near zero, we assume there must be poor indentation. In fact, the slope might be the aesthetic value of the indentation. A similar process could be developed for documentation and mnemonics.

7. CONCLUSIONS

We have seen how systematic changes in the style of C programs affect the fractal dimension in a statistically significant manner. Future research may consider the nature of these changes, i.e., how much beauty was added or removed by a change in style as suggested in the discussion. Another useful avenue is confirming these results for programming languages other than C.

REFERENCES

[1] Vermeulen, Allan & Ambler, Scott W., (2000) The Elements of Java Style, Cambridge

[2] Oulline, S., (1992) C Elements of Style: The Programmer's Style Manual for Elegant C and C++ Programs, M&T, 1992

[3] Google, Inc., (2015) "google-styleguide", http://code.google.com/p/google-styleguide/, accessed 11-May-2015

[4] NOAA National Weather Service, National Weather Service Office of Hydrologic Development, (2007) "General Software Development Standards and Guidelines Version 3.5"

[5] Kant, Immanuel, (1978) The Critique of Judgment (1790), translation by J. C. Meredith, Oxford University Press

[6] Free Software Foundation, (2015) http://linux.die.net/man/1/indent, access 13-May-2015

[7] Mandelbrot, Benoit, (1967) "How long is the coast of Britain? Statistical self-similarity and fractional dimension," Science, vol. 156 (3775), p. 636-638

[8] Peltgen, Heinz-Otto & Richter, P.H., (1986) The Beauty of Fractals, Springer, 1986

[9] Free Software Foundation (2015) http://www.gnu.org/software/coreutils/coreutils.html, accessed 11-May-2015
[10] Mandelbrot, Benoit, (1982) Fractal Geometry of Nature, Freeman, 1982
[11] Posnett, Daryl, Hindle, Abram & Devanbu, Prem, (2011) “A Simpler Model of Software Readability”, MSR ’11 Proceedings of the 8th Working Conference on Mining Software Repositories
[12] Buse, Raymond P.L., & Weimer, Westley R., (2008) “A metric for software readability,” ISSTA '08 Proceedings of the 2008 international symposium on Software testing and analysis
[13] Tran-Cao, De, Lévesque, Ghislain, & Meunier, Jean-Guy, (2004) "A Field Study of Software Functional Complexity Measurement," Proceedings of the 14th International Workshop on Software Measurement
[14] Oram, Andy & Wilson, Greg, eds. (2007) Beautiful Code, O’Reilly
[15] Gabriel, Richard, (1996) Patterns of Software, Oxford
[16] Schroeder, M., (2009) Fractals, Chaos, and Power Laws, Dover, 2009
[17] Cornforth, David, Jelinek, Herbert, Peichl, Leo, (2002) “Fractop: A Tool for Automated Biological Image Classification,” Proceedings of the Sixth Australia-Japan Joint Workshop on Intelligent and Evolutionary Systems, p. 1-8
[18] Gerl, Peter, Schönlieb, Carola, Wang, Kung Cheih, (2004) “The Use of Fractal Dimension in Arts Analysis,” Harmonic and Fractal Image Analysis, 2004, p. 70-73
[19] Coddington, Jim, Elton, John, & Rockmore, Daniel, Wang, Yang, (2008) “Multifractal analysis and authentication of Jackson Pollock paintings” Proc. SPIE 6810, Computer Image Analysis in the Study of Art, 68100F; doi: 10.1117/12.765015
[20] Taylor, R.P, Micolich, A.P., Jonas, D., (1999) “Fractal analysis of Pollock’s drip paintings,” Nature, vol. 399, June 1999
[21] Coleman, R, (2009) “Long-Memory of Pathfinding Aesthetics,” International Journal of Computer Games Technology, Volume 2009, Article ID 318505
[22] Coleman, R., (2009) “Fractal Analysis of Stealthy Pathfinding,” International Journal of Computer Games Technology, Special Issue on Artificial Intelligence for Computer Games, Volume 2009, Article ID 670459
[23] Coleman, R., (2008) “Fractal Analysis of Pathfinding Aesthetics,” International Journal of Simulation Modeling, Vol. 7, No. 2
[24] Coleman, Ron, Johnson, Matthew, (2014) ”A Study of Scala Repositories on Github”, International Journal of Advanced Computer Science Applications, vol. 5, issue 7, August 2014
[25] Coleman, Ron, Johnson, Matthew, (2014) “Power-Laws and Structure in Functional Programs,” Proceedings of 2014 International Conference on Computational Science & Computational Intelligence, Las Vegas, NV, IEEE Computer Society
[26] P. Kokol, J. Brest, and V. Zumer, “Long-range correlations in computer programs,” Cybernetics and systems, 28(1), 1997, p43-57
[27] P. Kokol, J. Brest, “Fractal structure of random programs,” SIGPLAN notices 33(6), 1998, p33-38
[28] P. Kokol “Searching for fractal structure in computer programs,” SIGPLAN 29(1), 1994
[29] Coleman, R., Pretty project, (2015) http://github.com/roncoleman125/Pretty, accessed 11-May-2015
[30] Conover, W.J., (1999) Practical Non-Parametric Statistics, Wiley

A NEW CRYPTOSYSTEM WITH FOUR LEVELS OF ENCRYPTION AND PARALLEL PROGRAMMING

Parag A. Guruji

Department of Computer Science and Engineering,
Walchand College of Engineering Sangli (An Autonomous Institute)
Vishrambag, Sangli, Maharashtra, India- 416415
gurujipa@gmail.com

ABSTRACT

Evolution in the communication systems has changed the paradigm of human life on this planet. The growing network facilities for the masses have converted this world to a village (or may be even smaller entity of human accommodation) in a sense that every part of the world is reachable for everyone in almost no time. But this fact is also not an exception for coins having two sides. With increasing use of communication networks the various threats to the privacy, integrity and confidentiality of the data sent over the network are also increasing, demanding the newer and newer security measures to be implied. The ancient techniques of coded messages are imitated in terms of new software environments under the domain of cryptography. The cryptosystems provide a means for the secured transmission of data over an unsecured channel by providing encoding and decoding functionalities. This paper proposes a new cryptosystem based on four levels of encryption. The system is suitable for communication within the trusted groups.

KEYWORDS

Matrix transformation, Fractionification, Re-integerization, Change of radix

1. INTRODUCTION

A cryptosystem refers to a suite of algorithms needed to implement a particular form of encryption and decryption. The encryption operations are the transformation functions with the set of all symbols which appear in data to be encrypted as their domain and the set of all corresponding encoded symbols as their codomain. The basic characteristic of any encryption operation for the faithful transmission of data is its reversibility. Any encryption operation that transforms input data into some encoded form must work as a bijective mapping, whose inverse exists and is also a bijective mapping. These criteria if not satisfied, the retrieval of the data from its encoded form back to its original form cannot be assured. Following figure represents the encryption operation f and its inverse f^{-1} (called decryption operation) as the bijective mappings from their corresponding domain and codomain.

David C. Wyld et al. (Eds) : CSEN, AISO, NCWC, SIPR - 2015
pp. 11–19, 2015. © CS & IT-CSCP 2015 DOI : 10.5121/csit.2015.51402

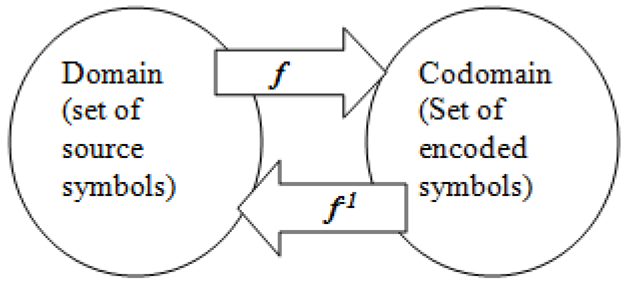


Figure 1. The encryption and decryption operations as mappings

In this paper the author proposes a new cryptosystem for the implementation in form of an application able to perform all the encryption and decryption tasks in an abstracted manner and thus keeping all of them transparent to only the valid user. The system operates on four levels of layers of the encryption making the complexity of cracking it extremely high. The four layers refer to the different set of operations, undergoes which the user data. The fragmentation and re-organization of the data is to be done as preprocessing before passing it to the encryption module. On the other side the decryption module works for the retrieval of encrypted data from the chunks that it receives and reorganizes it by sorting the randomly received chunks; after performing the four decryption operations on it which are inverses of the four encryption operations.

The encryption operations are:

1. Matrix transformation
2. Fractionification
3. Random no. addition
4. Change of radix

The decryption operations are:

1. Change of radix
2. Random no. removal
3. Re-integerization
4. Matrix re-transformation

The key generation operations are:

1. Random no. generation
2. Matrix generation using corresponding polynomial and checking for its inevitability
3. Radix generation using corresponding polynomial.

2. THE FUNCTIONAL DESIGN

2.1. Defining Tasks

To perform the operations in a systematic manner, the author defines the tasks to be performed on both the sides, the encryption and decryption as follows:

2.1.1. Encryption Tasks

1. Fragmentation of input data into chunks and indexing them.

2. Generation of the random key for each chunk and thus that of the key matrix and key radix.

3. Operating each chunk with encryption operations in the sequence in which they are listed above.

4. Augmentation of encrypted chunk with corresponding key which is a mere random integer.

2.1.2. Decryption Tasks

1. Receipt of the encrypted chunk and separation of key

2. Generation of the key matrix and key radix for received chunk.

3. Operating each chunk with decryption operations.

4. Reorganization of the chunks using the indices to retrieve data in its original form.

The selection/formation of polynomials required for the key generation are left on the implementation to keep this design flexible. The complexity of these polynomials will add to the complexity of whole of the system.

2.2. Task Accomplishment Scheme

The scheme for completing each of the above tasks is discussed in this section.

2.2.1. Fragmentation of input data into chunks and indexing:

The input data is fragmented in the chunks, each of size s bytes where s is the implementation-specific size defined for representation of an integer. The data structure to be used store these fragments is a linear list, each node of which contains a chunk and an index value representing the offset of that chunk from the beginning of the input data in terms of no. of chunks. Along with these two values, the chunk contains space for its key value, a random integer generated by the key generator. This fragmentation of the data enables the parallel functioning of every step to follow as discussed later in the paper.

2.2.2. Generation of the random key for each chunk and thus that of the key matrix and key radix:

A random number is to be generated (generation implementation specific) for each chunk and is then assigned as the key for that chunk. The selected polynomials are provided with this key to generate the key matrix and key radix for that chunk. The implementation must take care that the generated matrix will be an invertible (non singular) matrix. After completion of this step we are ready with required input values for the computation of the encrypted counterparts of each element in the input data.

2.2.3. Operating each chunk with encryption operations:

2.2.3.1. Matrix transformation:

The chunk formed along with the source file identifier (A random no. assigned to the source-file) and excluding the key is represented as a 3×1 matrix and is multiplied with the 3×3 matrix generated using the key (key matrix) to get the transformed matrix of order 3×1.

2.2.3.2. Fractionification:

The term Fractionification is defined as the conversion function which maps an integer to a fraction by dividing the integer by R^d where R is the radix of the number system under consideration and d is no. of significant digits in the original integer and then adding to it the integer value d. Thus, for an integer I in number system with radix R having d significant digits, fractionification f is defined as,

$$f(I) = I \div (R^d) + d$$

2.2.3.3. Random no. addition:

The fractionified no is then added with some random number multiplied by 10 to preserve the value of d (the no of significant digits in original no.). Thus, the integer I when fractionified and added with random no. becomes $r(I)$ given by,

$$r(I) = f(I) + n \times 10$$

where, n is the random number generated.

2.2.3.4. Change of radix:

Now that we have converted the integer I, representing s bytes of input data, to a floating point equivalent $r(I)$, the radix of the number system is to be changed as the outermost encoding operation. It is defined as the combination of two simple radix conversion operations, one for the integer part of the input floating point no. and other for its fraction part, represented as an integer. The target radix selection is important task and is selected using a randomization polynomial (implementation specific) with the key of corresponding chunk as its parameter. To use radix greater than 10, the corresponding symbols used are capital and small scripts of English alphabets and related numerical operations on them are to be defined.

2.2.4. Augmentation of encrypted chunk with corresponding key:

Once each element in the chunk except the key are encrypted, the chunk is augmented with the key, applied with fractionification and random no. addition, and thus is ready for the transmission.

2.2.5. Receipt of the encrypted chunk and separation of key:

The chunk when received on decryption end, it is to be stored in the buffer for unresolved chunks. From the key field the value of key is found and separated out and the corresponding key matrix and key radix are calculated exactly as explained above. The inverse of this matrix is calculated by adjoint method to get the decryption matrix.

2.2.6. Operating each chunk with decryption operations:

2.2.6.1. Change of radix:

Each element in the received chunk is operated on by the inverse change of radix with source radix as the one derived from the key and 10 as the target radix. Obviously, the integer and fraction part are treated individually treated as different integers, and then combined back.

2.2.6.2. Random no. removal:

Each element of the chunk is then operated upon by the inverse of the random no. addition to get the fractionified value using following function,

$$f(I) = (r(I)\%10)$$

2.2.6.3. Re-integerization:

The term Re-integerization is defined as the inverse function of Fractionification which maps to an integer, its equivalent fractionified value, and is defined as,

$$I = (f(I)\ \%1) \times 10^{d}$$

where,

$$d = \lfloor (f(I)\%10) \rfloor$$

2.2.6.4. Matrix re-transformation:

The chunk received is in the form of 3×1 matrix. It is multiplied by the 3×3 decryption matrix determined for that chunk according to the simple matrix multiplication to get the original data chunk.

2.2.7. Reorganization of the chunks

Now that having done with the decryption operations on received chunks, they are to be reorganized in the sequence of that of the data contained by them in the original source file. This is achieved by sorting the randomly placed data chunks using the identifier and index fields as the key. To boost the efficiency of sorting, author proposes to form a Binary Search Tree for each identifier and then the chunks are to be added in it according to the index field values as the key. Once the no. of nodes in the tree approaches to the total count of chunks present in the index field of identifier node, i.e. the root, the tree is traversed in In-order manner (left-root-right) and data field contents of each node are written into the destination file during traversal.

3. SECURITY FACTORS

The security and confidentiality of the data are the fundamental goals of any cryptosystem. In case of the proposed system, though all of such factors already have appeared in the discussion up till now, in this section we identify and enlist each of them for the getting the view of the security provided by the system as a whole.

1. Randomness of the key

2. Secrecy and complexity of the polynomials used for matrix and radix generation

3. Individual random key for each chunk: This removes the threat by many of the pattern analysis and known text attacks

4. Matrix Transformation: This transforms chunk into an integral unit whose meaning cannot be derived without accurate inverse of key matrix

5. Fractionification and Random no. addition: This covers the transformation and makes it too complex to analyze the resultant patterns and detect the transformation

6. Change of Radix: This changes the representation of the numbers and thus adding to the complexity of analysis of interrelations of elements in resultant values.

4. THE PARALLEL PROGRAMMING APPROACH

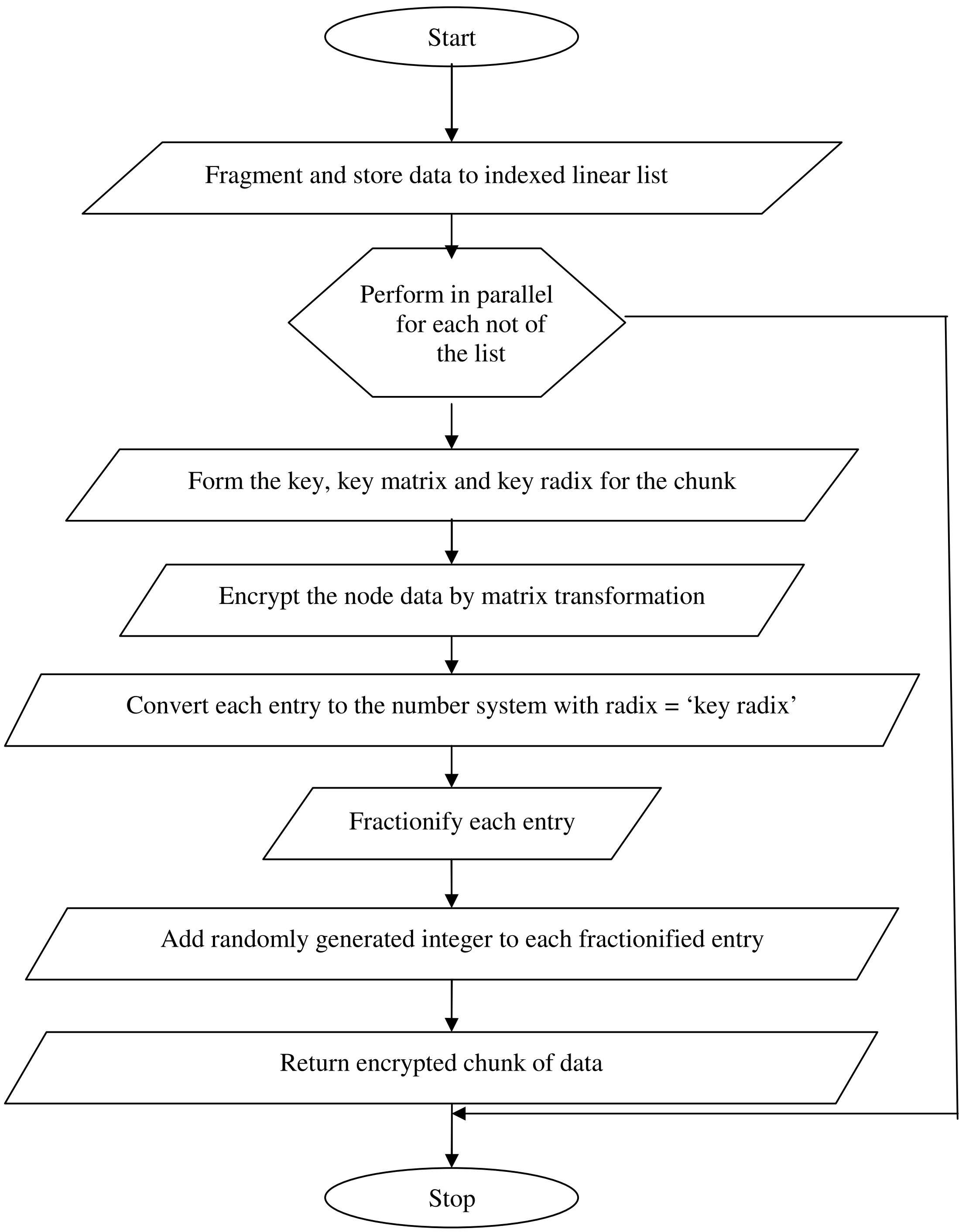
The important feature of proposed design of cryptosystem in this paper is the fragmentation of data and independency of the key for each fragment. This independence allows the parallel functioning of different modules in of cryptosystem. Each node follows the same path after the fragmentation is done. Thus after completion of Task 1on encryption side, each chunk is proposed to be processed in parallel through the completion of encryption. Also on decryption side, the received nodes are proposed to get processed in parallel till their addition to corresponding BST. This will reduce the time complexity of the cryptosystem application by the factor of n^{-1} where, n is the no. of fragments.

Along with this first level of parallel programming, the efficiency can further be increased by incorporating the second level of the same. In the second level, within each fragment the different elements are proposed to be operated with all the encryption as well as decryption functions independently except the matrix transformation. This will bring the time complexity to Time complexity of matrix multiplication + 3^{-1} (time complexity of rest of the operations with sequential approach), Thus reducing it roughly by factor of 3-1.

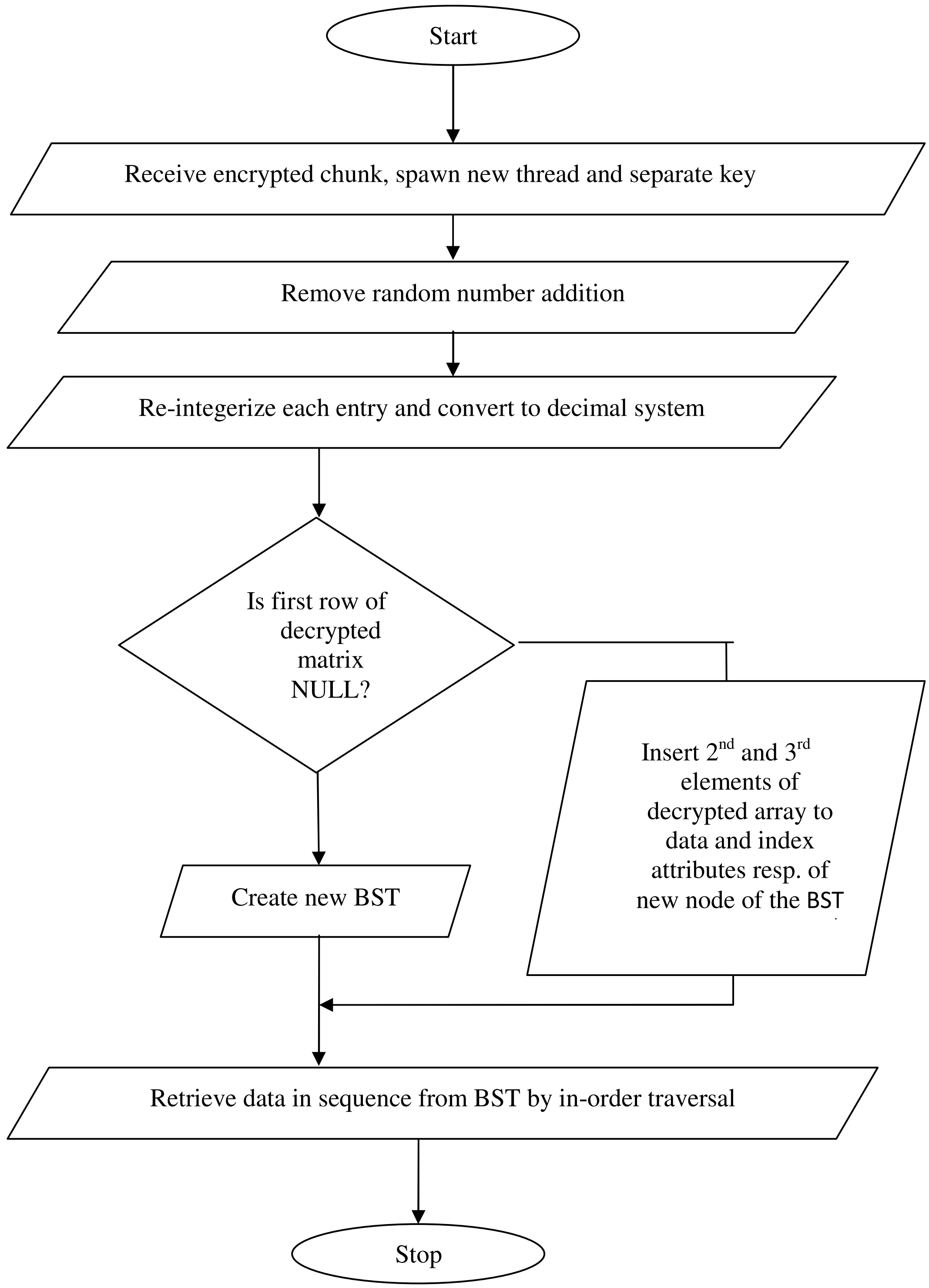
The author further proposes the third level of parallel programming, involving the parallel implementation of the matrix multiplication itself to further boost the efficiency. Thus the parallel programming approach adds to the efficiency significantly.

5. FLOWCHARTS

4.1. Encryption Flowchart:



4.2. Decryption Flowchart:



5. CONCLUSIONS

The cryptosystem proposed in the paper works on four different layers of the encryption. All the layers cover the possible attacks on its inner layer making the encryption extremely complex to crack. The security factors of the system protect it against the cracking attacks. The polynomials and random number generators are left to the implementation for making the system flexible. This incurs the variation of complexity of encryption depending on the implementation. The parallel approach of programming adds to the efficiency of application significantly, as discussed in the section II.

REFERENCES

[1] Yi-Shiung Yeh, Tzong –Chen Wu, Chin Chen Chang and Wei Chizh Yang "A New Cryptosystem using Matrix Transformation", Proceedings. 25th Annual IEEE International Carnahan Conference on Security Technology 1991 (Cat. No.91CH3031-2)

[2] D. C. Lay "Linear Algebra" ISBN: 9781405846219, Chapters 1, 2 and 3

AUTHORS

Parag A. Guruji

Earned Bachelor of Technology degree in Computer Science and Engineering from Walchand College of Engineering, Sangli, India in May 2014.

Working at ZLemma Analytics in Data Science team since June 2014

INTENTIONAL BLANK

DETECTION OF ORIENTED NUMBER PLATE IN VEHICLE USING AUTOCORRECTION FEATURE FROM GRAY LEVEL CO-OCCURENCE MATRIX

Veena M.N[1], Shruthi S.J[2] and Vasudev.T[2]

[1]P.E.T Research Foundation,
P.E.S. College of Engineering, Mandya, Karnataka, India
veenadisha1@gmail.com
[2]Maharaja Institute of Technology, Mysore, Karnataka, India
shruthijayaram131@gmail.com, vasu@mitmysore.in

ABSTRACT

The efficiency of an automatic number plate recognition system depends directly on the proper effective preprocessing of the number plate. The OCRs available for recognition are capable of reading the number plates which are in proper orientation of 00. In many situations the vehicle number plates captured may be in any different orientation like 900, 1800 and 2700. These orientations in number plates are due to declamping of number plate at one end or toppling of vehicle. Such differently oriented number plates cannot be subjected for reading by OCRs and such situations require the system to detect the direction of orientation and correct the same before subjecting the same for reading. This paper proposes a work to detect the orientation of segmented number plates from vehicle image using autocorrelation feature from gray level co-occurrence matrix. A good volume of training samples are generated synthetically to train the system and the system is tested using sufficient test samples. The results of system shows an overall efficiency of 65.61% and performs an essential preprocessing in an automatic number plate recognition system.

KEYWORDS

Number Plate, Orientation, Autocorrelation, OCRs

1. INTRODUCTION

Vehicle Number Plate Recognition(VNPR) system is an image processing application, developed to track the information about vehicles through the number plates. This application is gaining popularity in security and traffic monitoring systems. The number plate recognition system is important for variety of applications like automatic traffic congestion charge system, access control, tracing of stolen vehicles and identification of vehicles for traffic rules violations. The VNPR system plays a major role in automatic monitoring of traffic rules and maintains law enforcement on public roads. A VNPR system goes through a series of preprocessing task for

David C. Wyld et al. (Eds) : CSEN, AISO, NCWC, SIPR - 2015
pp. 21–27, 2015. © CS & IT-CSCP 2015 DOI : 10.5121/csit.2015.51403

efficient automatic reading. The generic OCRs available are designed to read the text in 0^0 orientations and are not capable of reading text in other orientations. The number plates segmented from captured vehicle images may not be in correct orientation i.e number plate instead of being at 0^0 orientation, it may be in any one of the 90^0, 180^0 and 270^0 orientations. The samples of segmented number plate in different orientation directions are shown in Figure1.

Figure. 1 (a) 0^0 Orientation (b) 90^0 Orientation (c) 180^0 Orientation (d) 270^0 Orientation

The different directions of orientations in number plates are due to unclamped number plates at one end or it may be due to toppled vehicles. Since OCRs do not to read such number plate oriented in other directions, the automatic number plate recognition (ANPR) system demands requirement of a preprocessing stage to detect the orientation of number plate and correct the same before subjecting for reading by OCRs. This requirement has motivated us to explore a method to detect the direction of orientation of the plate in vehicle images.

The work presented in this paper focuses on implementation of orientation detection in vehicle number plate. The rest of the paper is organized as follows section 2 gives brief survey about related research work. Section 3 presents the model designed for detection of orientation in vehicle number plate. Section 4 discuss about the experimental results and a brief conclusion is provided in section 5.

2. LITERATURE SURVEY

Many researchers have proposed good number of methods to detect tilt in the number plate of vehicles[1]. Tilt is the angular rotation noticed in the number plate. These methods are designed to detect the tilt which are less than 35^0 . These approaches are based on Principle Component Analysis [2], Fourier Transform[3], Hough Transform[4], Nearest neighbor connectors[5], Extreme points[6] and Moments[7,8]. But most approaches fail to detect the orientations of 90^0, 180^0 and 270^0. In literature sufficient quantum of works are reported on text skew detection in documents. The work in [9] proposed a modified approach of Hough transform for skew detection and correction of document images, the algorithm is computationally less efficient. The work in [10] proposed an integrated skew detection and correction using Fourier transform and DCT which is also computationally less efficient. Many different methodologies to detect skew in a given document page [11] were discussed. A traditional projection profile approach is proposed in [12]. In this approach features are extracted from each projection profile to determine the skew angle and it is quite cost effective. The improved nearest-neighbor based document skew detection method is proposed in [13] to estimate skew in documents respective to skew angle limitation. Another approach uses k-NN [14] clustering of the connected components. This approach has a relatively high accuracy but has a large computational cost, independent of the detection range. A skew detection method using the cross correlation between the text lines at a fixed distance [15] is based on the fact that the correlation between vertical lines in an image is maximum for a skewed document, is presented. It is found that the proposed method is computationally expensive and gives lesser accuracy. A bottom up technique for skew estimation based on nearest neighbor clustering is proposed in [16]. In this method, nearest neighbors of all the connected components are determined. Since only one nearest neighbor connectivity is made for each component, connection with noisy sub parts of characters would reduce the accuracy of the method. The above reported techniques have some limitations and depends on the factors like, speed, suitable only for text of sufficient size. Few techniques provide accurate results but slow in processing and other few techniques are cost effective but efficient in speed and accuracy. Some works are reported in literature to recognize the number plates in vehicle using size features of number plates[17]. But these approaches fail in distinguishing between number plate in 0^0 and 180^0 as well as between 90^0 and 270^0 for rectangle shaped number plates. In addition, the methods cannot be extended for square shaped number plates.

To the best of our knowledge, we could find a couple of works on detection of orientation in document images [18] and [19]. These methods are suitable for large text document and not ideal for number plate which contain very little text or very few characters. Hence a more suitable method is required to detect orientation of number plates and the work presents an attempt made in this direction.

3. PROPOSED MODEL

Input to the system is assumed to be the segmented number plate from the vehicle image. The sequence of stages in the work is shown in Figure.2. The process begins with necessary preprocessing to enhance the input number plate image for better character segmentation and character recognition. The preprocessed input image is subjected to detect the direction of orientation using autocorrelation feature from gray level co occurrence matrix (GLCM). Once the direction of the orientation is detected, next the same is corrected by rotating the segmented number plate in the counter direction. The subsequent subsections discuss the process in detail.

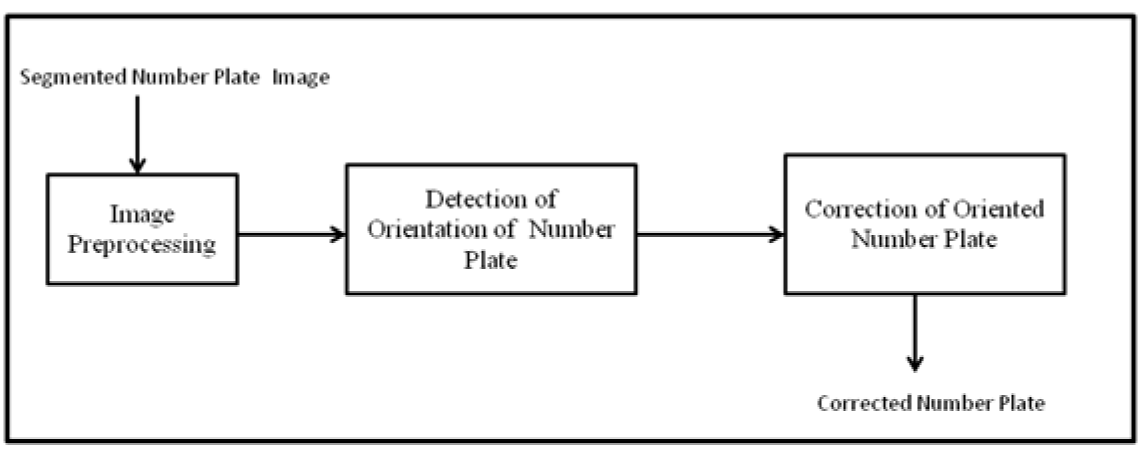


Figure 2. Stages in the Proposed Model

3.1. Preprocessing

The input image is converted into a grayscale image for easy analysis as it consists of only two color channels. The preliminarily preprocessing is carried out to remove noise using median filtering [20] from the input image. Median filter is a non-linear filter, which replaces the gray value of a pixel by the median of the gray values of its neighbours. A 3 × 3 mask is used to get eight neighbours of a pixel and their corresponding gray values. The gray value of the centre pixel of the mask is replaced by the median of the gray values of the pixels within the mask. This operation removes salt-and-pepper noise from the image. Figure.3 (a), (b) and (c) show the input, gray scale converted and filtered images respectively.

Figure 3. (a). Input Image (b). Gray Scale Image (c). Median Filtered Image

3.2. Orientation detection using Gray Level Co-variance Matrix (GLCM)

Texture is an important feature of an image. Texture plays an important role in many machine vision tasks such as surface inspection, scene classification, surface orientation and shape determination. Texture is characterized by the spatial distribution of gray levels in a neighbourhood. We can define texture as repeating patterns of local variations in image intensity which are to fine to be distinguished as separate objects at the observed resolution. Thus, a connected set of pixels satisfying a given gray-level property which occur repeatedly in an image region constitutes a textured region. To describe the texture of the region three approach are used in image processing these are statistical, structural and spectral. Statistical approaches specify the characterization of the textures by smooth, coarse, grainy, silky and so on. Since the texture is a spatial property, a simple one dimensional histogram is not useful in characterizing texture. In order to capture the spatial dependence of gray level values which contribute to the perception of texture, a two-dimensional dependence matrix known as gray level co occurrence matrix [21] is extensively used in texture analysis. Using a statistical approach such as co-occurrence matrix will help to provide valuable information about the relative position of the neighbouring pixels in an image. Another measure that has been used extensively in GLCM is the autocorrelation function[22] and this feature is mainly used for detection of orientation patterns in images.

3.2.1. Autocorrelation

The autocorrelation function [21] P(K, L) for an image 'f' is given by equation (1)

$$P(K,L)=\frac{\frac{1}{(N-K)(N-L)}\sum_{i=1}^{(N-K)}\sum_{j=1}^{(N-1)} f(i,j)f(i+k,j+l)}{1/N2\sum_{i=1}^{N}\sum_{j=1}^{N} f2(i,j)} \qquad 0\le K,l \le N-1 \qquad (1)$$

For images comprising repetitive texture pattern the autocorrelation function exhibits periodic behavior with a period equal to the spacing between adjacent texture primitives. When the texture is coarse, the autocorrelation function drop off slowly, where as for fine texture it drops off rapidly. The measure of periodicity of texture as well as a measure of the scale of the texture primitives. The autocorrelation range values for different orientations are computed and tabulated from the training samples. The Table.1 represents the Autocorrelation value ranges obtained from training samples for different orientations with limited overlaps.

Table 1 Autocorrelation value ranges

Orientation	Autocorrelation
0^0	>50
90^0	12 - 31
180^0	45 - 50
270^0	32 - 44

4. EXPERIMENTAL RESULTS

The experiments are conducted through testing 495 segmented vehicle number plate images obtained in different orientations. For each test sample the autocorrelation value is computed and the classification is made based on range values specified in Table 1. The Figures 4 (a), (b), (c),(d), (e), (f) and (g) shows few samples of oriented input images and corresponding corrected number plates.

Figure 4. (a) 0^0 Orientation (b). 90^0 Orientation (c). 180^0 Orientation (d).270^0 Orientation

(e). Corrected image of (b) (f). Corrected image of (c) (g). Corrected image of (d)

The results of the experiments are tabulated in table2. The result of overall efficiency of proposed method is 65.61% with 22.16% of wrong detections and 11.85% of rejections. The rejections are due to failures in autocorrelation threshold values. The wrong detections are due to overlapping of autocorrelation threshold values.

Table 2 Results of Detection Orientation in Vehicle Number Plate Image

Orientation	No. of Samples	Correct Detections	Wrong Detections	Rejections
0^0	100	75(75%)	17(17%)	8(8%)
90^0	145	89(61.38%)	36(24.82%)	20(13.79%)
180^0	115	70(60.87%)	30(26.09%)	15(13.04%)
270^0	135	88(65.18%)	28(20.74%)	17(12.59%)
Total	**495**	**322(65.61%)**	**111(22.16%)**	**60(11.85%)**

5. CONCLUSION

The work presented in this paper detects oriented number plates in vehicles which is an essential preprocessing required for an Vehicle Number Plate Recognition system under certain situations. The direction of orientation is detected using autocorrelation features from GLCM. The overall efficiency of proposed method is 65.61%. Misclassification is more between (0^0 and 180^0) and (90^0 and 270^0) due to overlapping of autocorrelation threshold values and rejections are due to computed in autocorrelation values does not fit in threshold ranges. Since it is an initial attempt made in this direction the success rate is relatively less. However there is much scope to minimize the misclassification using multi features of GLCM with multilevel classification which is under investigation.

REFERENCES

[1] Muhammad Kamal Hossen and Kaushik Deb " Vehicle License Plate Detection and Tilt Correction Based on HSI Color Moel and SUSAN Corner Detector ", Smart Computing Review, Vol.4 No. 5, 2014.

[2] Rajesh Bodade, Ram Bilas Pachori, Aakash Gupta, Pritesh Kanani, Deepak Yadav, "A Novel Approach for Automated Skew Correction of Vehicle Number Plate Using Principal Component Analysis" 2012 IEEE International Conference on Computational Intelligence and Computing Research.

[3] Lowther, S., Chandhran, V., Sridharan, S. An accurate method for skew determination in document images. In Proceedings of the Digital Image Computing Techniques and Applications. Melbourne (Australia), 2002.

[4] Arulmozhi, Arumuga Perumal , Thanooja Priyadarsini, Krishnan Nallaperumal, "Image Refinement Using Skew Angle Detection and Correction for Indian License Plates" 2012 IEEE International Conference on Computational Intelligence and Computing Research.

[5] Y. Lu, C. L. Tan., Improved Nearest Neighbor Based Approach to Accurate Document Skew Estimation, Department of Computer Science, School of Computing, National University of Singapore, Kent Ridge Singapore,2003.

[6] Marian Wagdy, Ibrahima Faye, Dayang Rohaya. Document Image Skew Detection and Correction Method Based on Extreme Points. In: 2014 IEEE.

[7] K. George ,K. Nicholas., A Fast High Precision Algorithm for the Estimation of Skew Angle Using Moments, Department of Informatics and Telecommunications University of Athens,2002.

[8] B. Darko , R. M. Dragan, An Algorithm for the Estimation of the Initial Text Skew, University of Belgrade, Technical Faculty Bor , Mining and Metallurgy Institute, Department of Informatics, Serbia, 2012.

[9] Deepak kumar, Dalwinder singh "modified approach of Hough transform for skew detection and correction in documented image" IJRCS,in 2012.

[10] Madir kaur, Simple Jindhal "an integrated skew detection and correction using fast fourier transform nd DCT",IJST,December 2013.
[11] R.singh and R kaur "skew detection in image processing", IJCTA, issn 2229-6093, vol 4,no.3(2013),pp.478-485.Postl, W., (1986).
[12] Detection of linear oblique structures and skew scan in digitized documents. In: Proc. 8th Internat. Conf. on Pattern Recognition, Paris, France, 687–689.
[13] M. Sarfraz, S.A. Mahmoud and Z. Rasheed,On Skew Estimation and Correction of Text, Computer Graphics, Imaging and Visualisation (CGIV 2007) 0-7695-2928-3/07.
[14] G.S.Lehal and Renu Dhir, (1999) "A Range Free Skew Detection Technique for Digitized Gurmukhi Script Documents", Proceedings 5th International Conference of Document Analysis and Recognition,Banglore,pp.147-152.
[15] Yan, H., "Skew correction of document images using interline cross correlation", Computer Vision, Graphics, and Image Processing 55,1993, pp 538-543.
[16] Hashizume, P. S. Yeh, A. Rosenfeld, "A method of detecting the orientation of aligned components", Pattern Recognition Letters, 1996,pp. 125-13
[17] Ahmed, M.Sarfaz, A. Zidouri and K.G AI-Khatib, " License plate recognition system", In the 10th IEEE International Conference on Electronics, Circuits and Systems, pp.898-901 (2003).
[18] Vasudev .T et.al. " Language Independent Skew Detection and Correction of Printed Text Document Images : A Non-rotational Approach, July 2006, VIVEK: Journal of Artificial Intelliegence, Vol.16, No.10,pp.8-15,2006.
[19] Vasudev .T, et.al " Detection and Correction of Page Orientation in Monochrome Textual Document Image" Journal of Computer Science, Vol. 2, No.1,PP.465-473, Jan-2008.
[20] Gonzalez .R.C, R.E .Woods ,"Digital Image Processing", 2nd Edition, Printice Hall, Englewood Cliffs , NY 2002
[21] Andrik Rampun, Harry Strange, " Texture Segmentation using different orientations of GLCM features", MIRAGE'13, June 06-07-2013, Berlin, Germany.
[22] Ramesh Jain, Rangachar Kastrui " Machine Vision" Published by Mc.Graw-Hill, Inc, ISBN 0-07-0320187,1995.

AUTHORS

Veena M.N, obtained her Master of Computer Applications(MCA) from University of Mysore in 1998. She received her one more Master degree in Computer science and technology from University of Mysore in 2007.She is having 15 years of experience in academics and her area of research is Digital Image Processing and Computer Vision. She is pursuing doctoral degree under University of Mysore.

Shruthi S.J, Obtained her Bachelor degree in Information Science and Engineering from Ghousia College of Engineering, Ramanagara in 2013. She is currently pursuing Master degree in Computer Science and Engineering under VTU, Belgaum.

Vasudev .T is Professor, in the Department of Master of Computer Applications, Maharaja Institute of Technology, Mysore. He obtained his Bachelor of Science and post graduate diploma in computer programming with two Masters Degrees one in Computer Applications and other one is Computer Science and Technology. He was awarded PhD. in Computer Science from University of Mysore. He is having 30 years of experience in academics and his area of research is Digital Image Processing specifically document image processing.

INTENTIONAL BLANK

A SYSTEM FOR DETECTION OF FABRICATION IN PHOTOCOPY DOCUMENT

Suman V Patgar, Vasudev T and Murali S

Maharaja Research Foundation,
Maharaja Institute of Technology Mysore,
Belawadi, S.R Patna, Mandya, India, 571438
sumanpatgar@gmail.com

ABSTRACT

Photocopy documents are very common in our normal life. In India, people are permitted to carry and produce photocopied documents frequently, to avoid damages or losing the original documents. But this provision is misused for temporary benefits by fabricating fake photocopied documents. When a photocopied document is produced, it may be required to check for its originality. An attempt is made in this direction to detect such fabricated photocopied documents. This paper proposes an unsupervised system to detect fabrication in photocopied document. The work in this paper mainly focuses on detection of fabrication in photocopied documents in which some contents are manipulated by new contents above it using different ways. Testing is performed with a different set of collected testing samples resulted in an average detection rate of 70.14%.

KEYWORDS

Document, Photocopy, Fabrication, Unsupervised system

1. INTRODUCTION

Many authorities in India trust and consider the photocopied documents submitted by citizens as proof and accept the same as genuine. Few such applications like to open bank account, applying for gas connection, requesting for mobile sim card, the concerned authorities insist photocopy documents like voter id, driving license, ration card, pan card and passport as proof of address, age, photo id etc to be submitted along with the application form. Certain class of people could exploit the trust of such authorities, and indulge in forging/ tampering/ fabricating photocopy document for short / long term benefits unlawfully. Fabricated photocopy is generated normally by making required changes intelligently in the photocopy obtained from an original document and then taking the recursive photocopy [1] from the modified photocopy. It is learned that in majority cases, fabrications are made by replacing a different photograph in place of photograph of an authenticated person; replacing contents in variable regions [2] through cut-and-paste technique from one or more documents; overlaying new content above actual content; adding new content into existing content; removing some content from existing; changing content by

David C. Wyld et al. (Eds) : CSEN, AISO, NCWC, SIPR - 2015
pp. 29–35, 2015. © CS & IT-CSCP 2015 DOI : 10.5121/csit.2015.51404

overwriting; intellectually changing character in contents. It is quite evident from the applications listed above, the fabrication could be mainly made in the variable regions of documents.

The fabricated photocopy documents are generated to gain some short term or long term benefits unlawfully. This poses a serious threat to the system and the economics of a nation. In general, such frauds are noticed in the application areas where photocopy documents are just enough. These types of systems trusting photocopied document raise an alarm to have an expert system [3] that efficiently supports in detecting a forged photocopy document. The need of such requirement to the society has motivated us to take up research through investigating different approaches to detect fabrication in photocopy document.

2. RELATED WORKS

In literature no significant effort is noticed towards research on fabricated photocopy documents except for the work of detection of fabrication in photocopy document using GLCM features [4]. In this work, attempt is made to detect fabrication of photocopy documents in which text in variable region is fabricated by putting new contents in many ways. Many research attempts are carried out on original documents instead on photocopied documents, like signature verification, detection of forged signature [5], handwriting forgery [6], printed data forgery [7], and finding authenticity of printed security documents [8]. Literature survey in this direction reveals that the above research attempts have been made in the following issues: Discriminating duplicate cheques from genuine ones [8] using Non-linear kernel function; Detecting counterfeit or manipulation of printed document [7] and this work is extended to classify laser and inkjet printouts; Recognition and verification of currency notes of different countries [9] using society of neural networks along with a small work addressing on forged currencies; Identification of forged handwriting [6] using wrinkles as a feature is attempted along with comparison of genuine handwriting. The domain of research is in its early stage and there is no standard data set available for experimentation. The fabricated samples are obtained through writing new contents over an actual contents, smeared whitener, cut and paste and adding new contents above it.

The organization of the paper is as follows. In Section 3, the proposed model is explained with the support of a block diagram along with a brief introduction. The experimental results and a qualitative comparative analysis on the state of the art techniques are discussed in Section 4. The work is concluded in Section 5.

3. PROPOSED MODEL

Figure 1 shows the flow sequence of stages in the fabrication detection process. The system takes the photocopy document as input. The photocopy documents are the one which are subjected for detection of fabrication in textual region. The input photocopy document is scanned at 300 dpi to obtain bitmap image. The image is preprocessed to minimize the noise using median filter. The next stage considers the image and segment variable regions from the document image for detection of fabrication in the segmented area [12]**.** The segmented part is used to detect fabrication in the variable region [4,10,13].

As the area/ region of fabrication in the document is relatively smaller, compared to entire document it is quite hard to take decision whether the document is fabricated or non-fabricated. Hence it is favorable to employ all the three methods to check for the fabrication [4,10,13]. The content in the variable region is decided as fabricated when one or more of the approaches lead to the classification as fabricated class. The content in the variable region is decided as non fabricated when all the three approaches confirm the non fabrication. The rejection cases in the method are considered as conflict and suggested for manual verification.

It is quite clear from the Figure 1, estimation of recursive order number of the photocopy is an input to the validation stage [1]. When the detection stage indicates the photocopy is not fabricated and the recursive order number is 1 then it is confirmed that the photocopy is not fabricated and the photocopy is considered as genuine photocopy. When the detection stage indicates fabrication in photocopy and the recursive order number is 2 or above then it can be confirmed that the photocopy is fabricated. In such case, the photocopy submitted can be rejected. Remaining combinations are the conflict cases and forwarded for manual verification with the original document at a later point in time. The decision rule for validation is illustrated in Figure 2 in the form of decision tree.

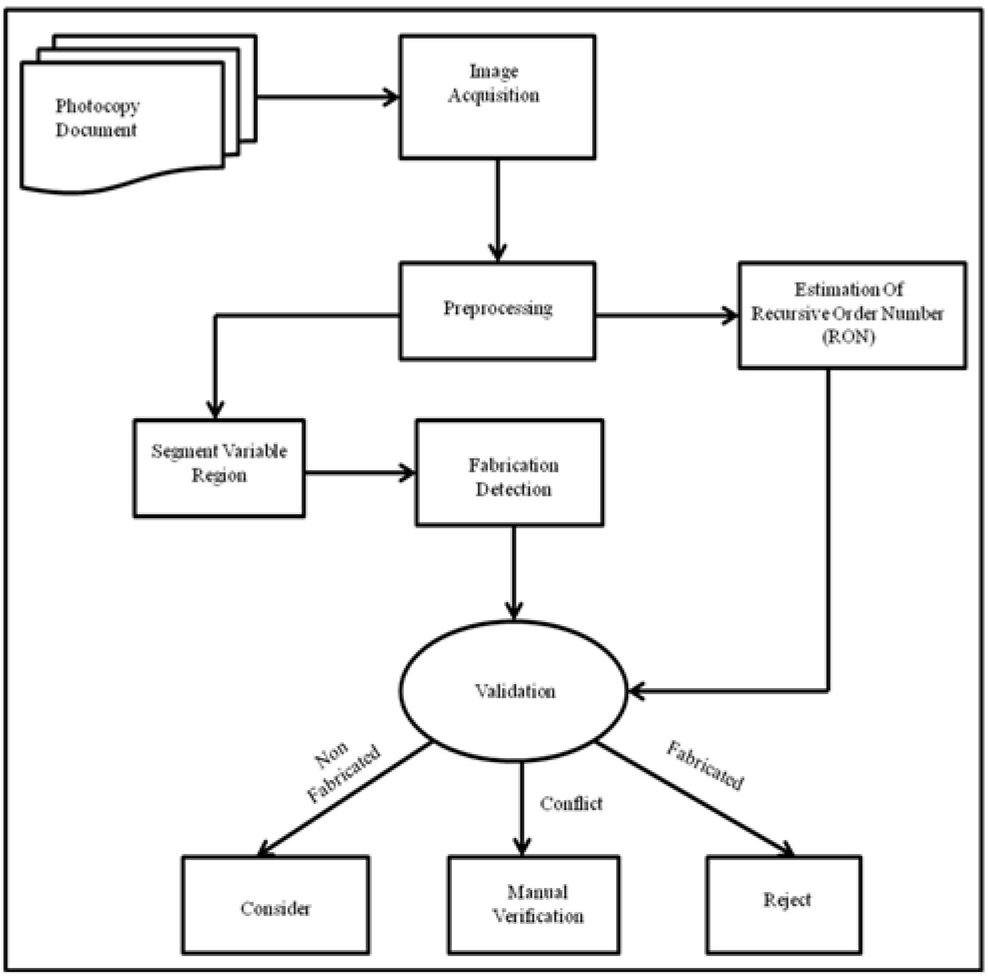


Figure 1: Flow sequence of stages in Detection of Fabrication

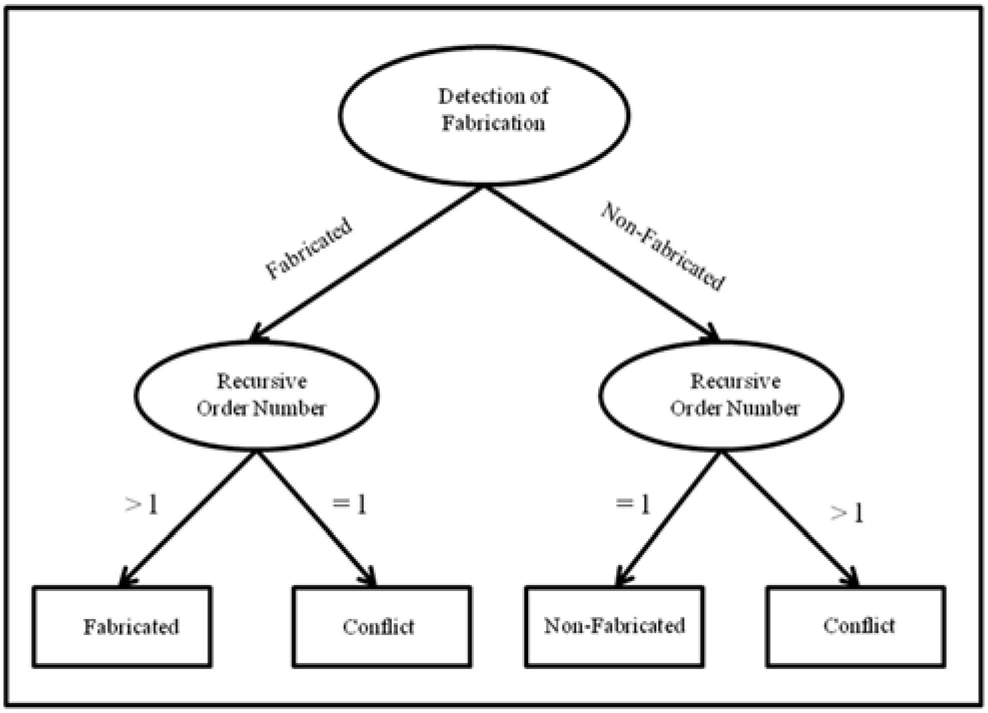


Figure 2: Decision Tree for Validation

4. EXPERIMENTAL RESULT

The proposed model is tested through considering 134 numbers of samples consisting of fabricated and non fabricated photocopy documents generated synthetically. The samples include DDs, certificates, birth / death certificates, cheques, attendance certificates, participation certificate and degree certificates obtained at different recursive orders. Samples are selected on which variable regions are underlined. Table (1) through (3) show results obtained at each stage.

Table 1 illustrates the results of 1st stage. This step shows correct segmentation efficiency as 82.83%. Table 1 indicates that 23 out of 134 samples are incorrect segmentation. The system is very much sensitive for improper segmentation. Every segmented region is forwarded to the detection of fabrication stage.

Table 1: Segmentation of Variable Region Result

No. of Samples	**Correct Segmented**	**Incorrect Segmented**	**Segmentation Efficiency**
134	111	23	82.83%

Table 2 shows the results of the fabrication detection stage. The segmented regions of the previous stage are the input for this stage.

Table 2: Detection of Fabrication Result

No. of samples	**No. of correct classified**	**No. of Misclassified**	**Classification Efficiency**
134	98(73.13%)	36(26.87%)	73.13%

The validation stage attempts to confirm the decision of fabrication detection stage and re-tunes the misclassification through classifying them as to correct class or conflict class. The validation stage considers fabrication detection output and recursive order number of the photocopy as input for validation. The Table 3 shows the result of detection of fabrication after validation.

Table 3: Validation Result

No. of samples	**No. of correct classified**	**No. of Conflicts**	**Validation Efficiency**
134	94(70.14%)	40(29.86%)	70.14%

The system is very much sensitive for improper segmentation. Because of the incorrect segmentation in input a considerable amount of drop in efficiency is noticed in this stage.

5. CONCLUSION

A complete integrated system has been proposed in this paper to detect fabrication in photocopy documents. The system takes the photocopy document as input and goes through various stages for classification of photocopy document. An overall performance of the system is about 70.14%. The reduction of efficiency in the system is due to efficiency drop happening at each stage in the system. The domain of the problem is new and challenging to come out with more generic solution especially in the stage of segmentation of variable region in the document. Scope to explore more efficient models to investigate fabrication in variable region is still open for research. The main limitation in the system is failures in estimating lower order recursive order number. More accurate models are required to detect lower recursive order number. More perfection is required in segmentation of variable region from documents. The automatic detection model has certain limitations like failure in detecting fabrication in the minute change in first order photocopy document and removing or adding few leading or trailing characters. There is scope to improve the efficiency of detection of fabrication by strengthening the model through improving segmentation of variable region in photocopy document.

REFERENCES

[1] Suman Patgar, Vasudev T, Estimation of Recursive Order Number of a Photocopied Document Through Entropy From Gray Level Co-occurrence Matrix, 2013.

[2] Vasudev T, Automatic Data Extraction from Pre-Printed Input Data Forms: Some New Approaches, PhD thesis supervised by Dr. G. Hemanthakumar, University of Mysore, India, 2007.

[3] Rich Kevin Knight, Artificial Intelligence, 2nd Edition, McGraw-Hill Higher Education.

[4] Suman Patgar, Vasudev T, 2013 An Unsupervised Intelligent System to Detect Fabrication in Photocopy Document Using Geometric Moments and Gray Level Co-Occurrence Matrix, IJCA(0975-8887) volume- 74/N0. 12, July 2013.

[5] Madasu Hanmandlu, Mohd. Hafizuddin Mohd. Yusof, Vamsi Krishna Madasu off-line signature verification and forgery detection using fuzzy modeling Pattern Recognition Vol. 38, pp 341-356, 2005.

[6] Cha, S.-H., & Tapert, C. C., Automatic Detection of Handwriting forgery, Proc. 8thInt.Workshop Frontiers Handwriting Recognition(IWFHR-8), Niagara, Canada, pp 264-267, 2002.

[7] Christoph H Lampert, Lin Mei, Thomas M Breuel Printing Technique Classification for Document Counterfeit Detection Computational Intelligence and Security, International Conference, Vol. 1, pp 639-644, 2006.

[8] Utpal Garian, Biswajith Halder, On Automatic Authenticity Verification of Printed Security Documents, IEEE Computer Society Sixth Indian Conference on Computer vision, Graphics & Image Processing, pp 706-713, 2008.

[9] Angelo Frosini, Marco Gori, Paolo Priami, A Neural Network-Based Model For paper Currency Recognition and Verification IEEE Transactions on Neural Networks, Vol. 7, No. 6, Nov 1996.

[10] Suman V Patgar, Y H Sharath Kumar, Vasudev T, Detection of Fabrication in Photocopy Document Using Texture Features through K-Means Clustering, Signal & Image Processing : Signal & Image Processing: An International Journal (SIPIJ), Vol.5, No.4, pp: 29-36, August 2014.

[11] Suman V Patgar, Rani K, Vasudev T, A Voting based Approach to Detect Recursive Order Number of Photocopy Documents using Probability Distributions, Signal & Image Processing : An International Journal(SIPIJ), Vol.5, No.4, pp: 37-44, August 2014.

[12] Suman V Patgar, Y H Sharath Kumar, Vasudev T, Segmentation of Handwritten Text from Underlined Variable Regions in Documents, International Journal of Information Processing(IJIP), Volume 8, issue 3, pp:23-30, 2014

[13] Suman V Patgar, Rani K, Vasudev T, An unsupervised intelligent system to detect fabrication in photocopy document using Variations in Bounding Box Features, accepted in International conference on Contemporary Computing and Informatics (IC3I), 2014, Mysore, India

AUTHORS

Dr. Vasudev T is Professor, in the Department of Computer Applications, Maharaja Institute of Technology, Mysore. He obtained his Bachelor of Science and post graduate diploma in computer programming with two Masters Degrees one in Computer Applications and other one is Computer Science and Technology. He was awarded Ph.D. in Computer Science from University of Mysore. He is having 30 years of experience in academics and his area of research is Digital Image Processing specifically document image processing and authored about 50 research papers.

Dr. Mural S has obtained his Bachelors, Masters and PhD degrees from University of Mysore. Presently he is working as Professor in the Department Computer Science and Engineering, Maharaja Institute of Technology, Mysore, Karnataka, India. He has a total experience of 30+ years expereince in teaching and about 20 years of experience in research. His area of research is Computer Vision and authored nearly 200 research papers and articles. He has served at different capacities as member and chairman in various academic bodies of Universities, UGC and AICTE. He is the Founder President of Maharaja Educational Trust, Mysore having around 10 different academic institutions.

Dr. Suman V Patgar, obtained her Bachelor of Engineering from Kuvempu University in 1998. She received her Masters degree in Computer Science and Engineering from VTU Belgaum in 2004. She was awarded Ph.D. in Computer Science from University of Mysore in 2015. She is having 10 years of experience in academics and her area of research is Digital Image Processing specifically document image processing and authored about 10 research papers.

INTENTIONAL BLANK

X-TREPAN: A MULTI CLASS REGRESSION AND ADAPTED EXTRACTION OF COMPREHENSIBLE DECISION TREE IN ARTIFICIAL NEURAL NETWORKS

Awudu Karim and Shangbo Zhou

College of Computer Science, Chongqing University,
Chongqing, 400030, China
awudubody@yahoo.com

ABSTRACT

In this work, the TREPAN algorithm is enhanced and extended for extracting decision trees from neural networks. We empirically evaluated the performance of the algorithm on a set of databases from real world events. This benchmark enhancement was achieved by adapting Single-test TREPAN and C4.5 decision tree induction algorithms to analyze the datasets. The models are then compared with X-TREPAN for comprehensibility and classification accuracy. Furthermore, we validate the experimentations by applying statistical methods. Finally, the modified algorithm is extended to work with multi-class regression problems and the ability to comprehend generalized feed forward networks is achieved.

KEYWORDS

Neural Network, Feed Forward, Decision Tree, Extraction, Classification, Comprehensibility.

1. INTRODUCTION

Artificial neural networks are modeled based on the human brain architecture. They offer a means of efficiently modeling large and complex problems in which there are hundreds of independent variables that have many interactions. Neural networks generate their own implicit rules by learning from examples. Artificial neural networks have been applied to a variety of problem domains [1] such as medical diagnostics [2], games [3], robotics [4], speech generation [5] and speech recognition [6]. The generalization ability of neural networks has proved to be superior to other learning systems over a wide range of applications [7].

However despite their relative success, the further adoption of neural networks in some areas has been impeded due to their inability to explain, in a comprehensible form, how a decision has been arrived at. This lack of transparency in the neural network's reasoning has been termed the Black Box problem. Andrews et al. [8] observed that ANNs must obtain the capability to explain their decision in a human-comprehensible form before they can gain widespread acceptance and to

David C. Wyld et al. (Eds) : CSEN, AISO, NCWC, SIPR - 2015
pp. 37–54, 2015. © CS & IT-CSCP 2015 DOI : 10.5121/csit.2015.51405

enhance their overall utility as learning and generalization tools. This work intends to enhance TREPAN to be able to handle not only multi-class classification type but also multi-class regression type problems. And also to demonstrate that X-TREPAN can understand and analyze generalized feed forward networks (GFF). TREPAN is tested on different datasets and best settings for TREPAN algorithm are explored based on database type to generate heuristics for various problem domains. The best TREPAN model is then compared to the baseline C4.5 decision tree algorithm to test for accuracy.

Neural networks store their "Knowledge" in a series of real-valued weight matrices representing a combination of nonlinear transforms from an input space to an output space. Rule extraction attempts to translate this numerically stored knowledge into a symbolic form that can be readily comprehended. The ability to extract symbolic knowledge has many potential advantages: the knowledge obtained from the neural network can lead to new insights into patterns and dependencies within the data; from symbolic knowledge, it is easier to see which features of the data are the most important; and the explanation of a decision is essential for many applications, such as safety critical systems. Andrews et al. and Ticke et al. [9], [10] summarize several proposed approaches to rule extraction. Many of the earlier approaches required a specialized neural network architectures or training schemes. This limited their applicability; in particular they cannot be applied to in situ neural networks. The other approach is to view the extraction process as learning task. This approach does not examine the weight matrices directly but tries to approximate the neural network by learning its input-output mappings. Decision trees are a graphical representation of a decision process. The combination of symbolic information and graphical presentation make decision trees one of the most comprehensible representations of pattern recognition knowledge.

2. BACKGROUND AND LITERATURE REVIEW

2.1 Artificial Neural Network

Artificial neural networks as the name implies are modeled on the architecture of the human brain. They offer a means of efficiently modeling large and complex problems in which there may be hundreds of independent variables that have many interactions. Neural networks learn from examples by generating their own implicit rules. The generalization ability of neural networks has proved to be equal or superior to other learning systems over a wide range of applications.

2.2 Neural Network Architecture

A neural network consists of a large number of units called processing elements or nodes or neurons that are connected on a parallel scale. The network starts with an input layer, where each node corresponds to an independent variable. Input nodes are connected to a number of nodes in a hidden layer. There may be more than one hidden layer and an output layer. Each node in the hidden layer takes in a set of inputs (X1, X2, …, Xm), multiplies them by a connection weight (W1, W2, …, Wm), then applies a function, f(WTX) to them and then passes the output to the nodes of the next layer. The connection weights are the unknown parameters that are estimated by an iterative training method to indicate the connection's strength and excitation. The calculation of the final outputs of the network proceeds layer by layer [11]. Each processing element of the hidden layer computes its output as a function of linear combination of inputs from the previous layer plus a bias. This output is propagated as input to the next layer and so on until the final layer is reached. Figure 1 shows the model of a single neuron [12]

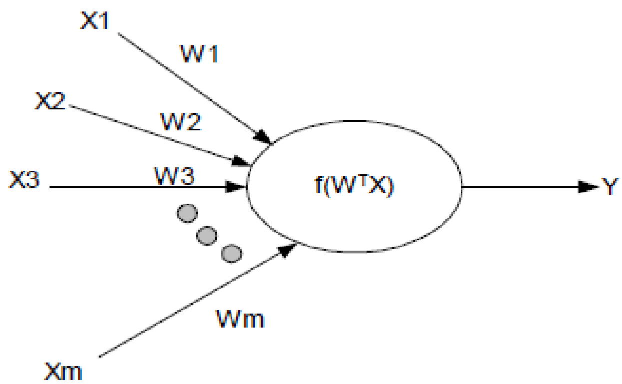


Figure 1. Model of a Single Neuron

The output of the neuron can be expressed as

$$Y = f(\sum_{i=1}^{m} w_i x_i)\text{, or}$$

$$Y = f(W^T X)$$

In the above equations, W is the weight vector of the neural node, defined as

$W = [w_1, w_2, w_3, \ldots\ldots\ldots\ldots w_m]^T$ and X is the input vector , defined as

$$X = [x_1, x_2, x_3, \ldots\ldots\ldots\ldots x_m]^T$$

Figure 2. Shows a typical neural network architecture representation.

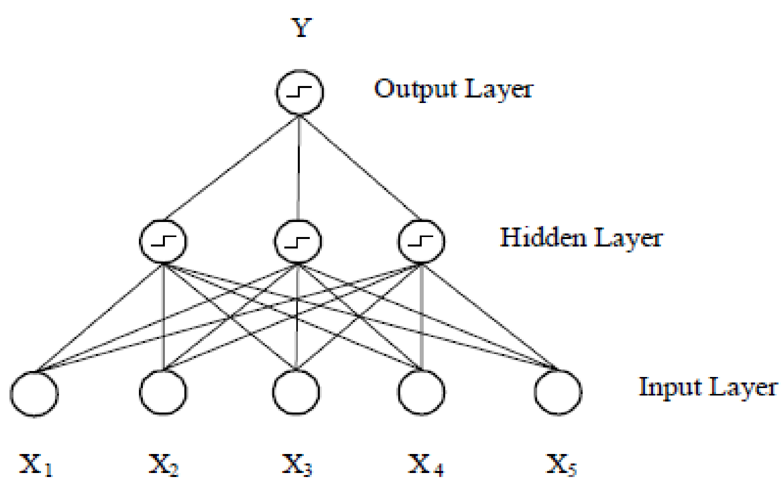


Figure 2. Neural Network Architecture

These are different types of activation functions that can be applied at the node of the network. Two of the most commonly used neural network functions are the hyperbolic and logistic (or sigmoid) functions. They are sometimes referred to as "squashing" functions since they map the inputs into a bounded range. Table 1 shows a list of activation functions that are available for use in neural networks.

Table 1. Activation Functions used in Neural Networks Adapted from [13]

Functions	**Definition**	**Range**
Identity	x	$(-\infty,+\infty)$
Logistic	$\frac{1}{(1-e^{-x})}$	$(0,+1)$
Hyperbolic	$\frac{e^{x}-e^{-x}}{e^{x}+e^{-x}}$	$(-1,+1)$
Exponential	e^{x}	$(0,+\infty)$
Softmax	$\frac{e^{-x}}{\sum_i e^{x_i}}$	$(0,+1)$
Unit Sum	$\frac{x}{\sum_i x_i}$	$(0,+1)$
Square root	$\sqrt{x}$	$(0,+\infty)$
Sine	$Sin(x)$	$(0,+1)$
Ramp	$\begin{cases} -1, x \leq -1 \\ x, -1 < x \\ +1, x \geq +1 \end{cases}$	$(-1,+1)$
Step	$\begin{cases} 0, x < 0 \\ +1, x \geq 0 \end{cases}$	$(0,+1)$

2.3 Multilayer Perceptrons

Multilayer Perceptrons (MLOs) are layered feed forward networks typically trained with back propagation. These networks have been used in numerous applications. Their main advantage is that they are easy to use, and that they can approximate any input/output map. A major disadvantage is that they train slowly, require lots of training data (typically three times more training samples then network weights)[14].

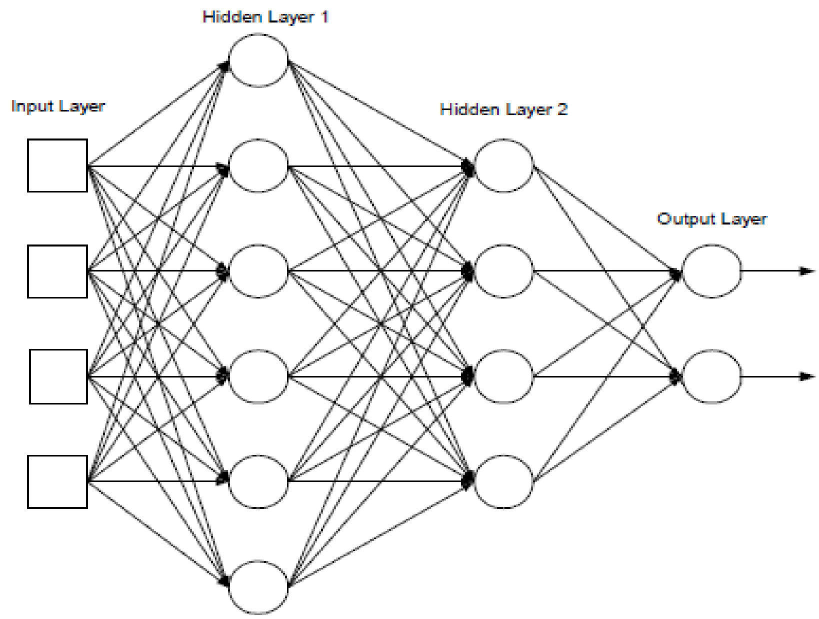


Figure 3. A schematic Multilayered Perceptron Network

A Generalized Feed Forward (GFF) network is a special case of a Multilayer Perception wherein connections can jump over one or more layers. Although an MLP can solve any problem that a GFF can solve, in practice, a GFF network can solve the problem more efficiently [14]. Figure 4 shows a general schematic of a Generalized Feed Forward Network.

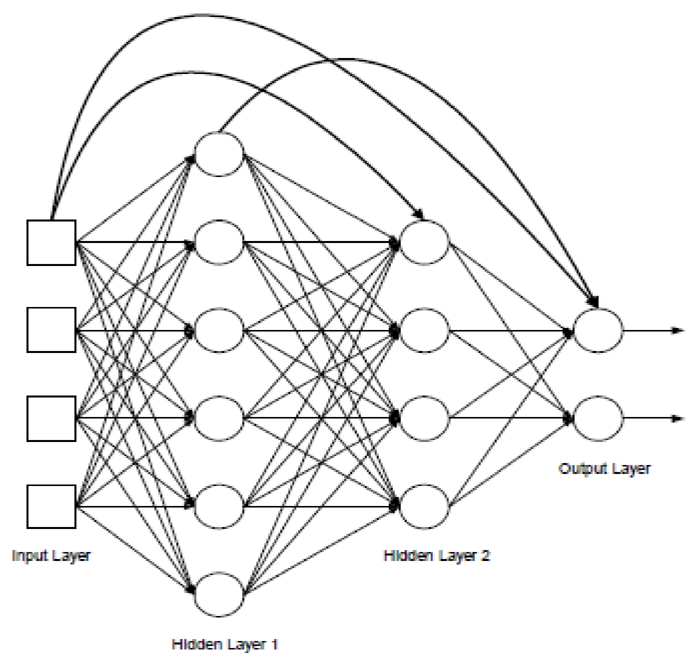


Figure 4. Generalized Feed Forward Networks

2.4 Neural Networks for Classification and Regression

Neural networks are one of the most widely used algorithms for classification problems. The output layer is indicative of the decision of the classifier. The cross entropy error function is most commonly used in classification problems in combination with logistic or soft max activation functions. Cross entropy assumes that the probability of the predicated values in a classification problem lie between 0 and 1. In a classification problem each output node of a neural network represents a different hypothesis and the node activations represent the probability that each hypothesis may be true. Each output node represents a probability distribution and the cross entropy measures calculate the difference between the network distribution and the actual distribution [15]. Assigning credit risk (good or bad) is an example of a neural network classification problem. Regression involves prediction the values of a continuous variable based on previously collected data. Mean square error is the function used for computing the error in regression networks. Projecting the profit of a company based on previous year's data is regression type neural network problem.

2.5 Neural Network Training

The neural network approach is a two stage process. In the first stage a generalized network that maps the inputs data to the desired output using a training algorithm is derived. The next stage is the "production" phase where the network is tested for its generalization ability against a new set of data.

Often the neural network tends to over train and memorizes the data. To avoid this possibility, a cross-validation data set is use. The cross validation data set is a part of the data set which is set aside before training and is used to determine the level of generalization produced by the training set. As training processes the training error drops progressively. At first the cross validation error decreases but then begins to rise as the network over trains. Best generalization ability of the network can be tapped by stopping the algorithm where the error on the cross validation set starts to rise. Figure 5 illustrates the use of cross-validation during training.

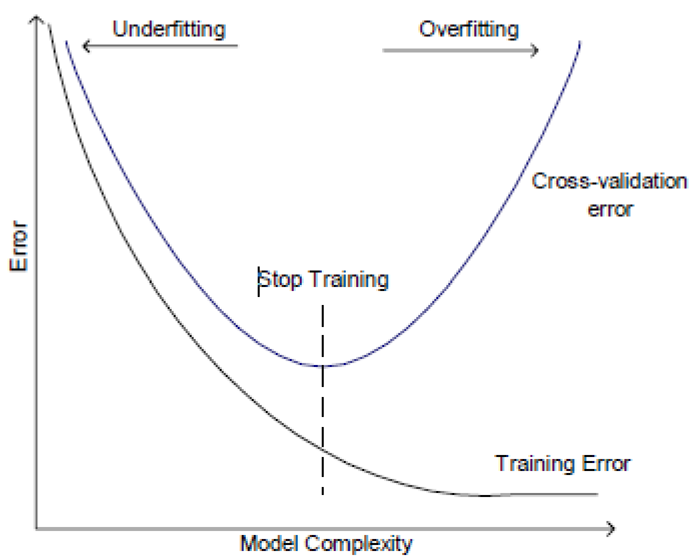


Figure 5. Use of cross-Validation during Training

2.6 Rule Extraction from Neural Networks

Although neural networks are known to be robust classifiers, they have found limited use in decision-critical applications such as medical systems. Trained neural networks act like black boxes and are often difficult to interpret [16]. The availability of a system that would provide an explanation of the input/output mappings of a neural network in the form of rules would thus be very useful. Rule extraction is one such system that tries to elucidate to the user, how the neural network arrived at its decision in the form of if-then rules.

Two explicit approaches have been defined to date for transforming the knowledge and weights contained in a neural network into a set of symbolic rules de-compositional and pedagogical [17]. In the de-compositional approach the focus is on the extracting rules at an individual hidden and/or output level into a binary outcome. It involves the analysis of the weight vectors and biases associated with the processing elements in general. The subset [18] algorithm is an example of this category. The pedagogical approach treats neural networks like black boxes and aims to extract rules that map inputs directly to its output. The Validity Interval Analysis (VIA) [19] proposed by Thrum and TREPAN [20] is an example of one such technique .Andrews et al [21] proposed a third category called eclectic which combines the elements of the basic categories.

2.7 Decision Trees

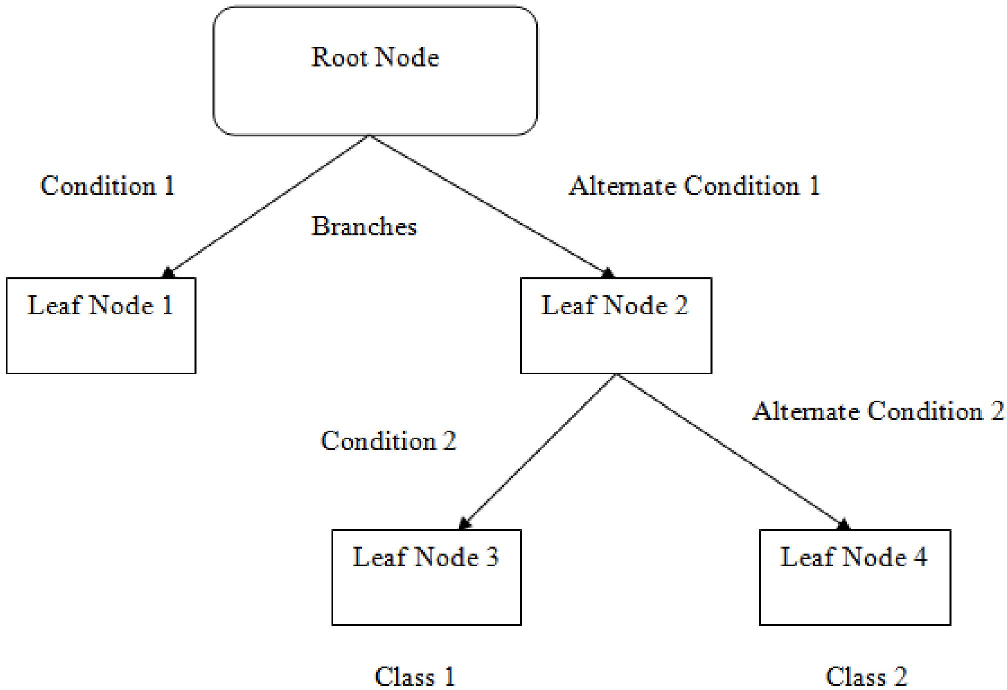
A decision tree is a special type of graph drawn in the form of a tree structure. It consists of internal nodes each associated with a logical test and its possible consequences. Decision trees are probably the most widely used symbolic learning algorithms as are neural networks in the non-symbolic category.

2.8 Decision Tree Classification

Decision trees classify data through recursive partitioning of the data set into mutually exclusive subsets which best explain the variation in the dependent variable under observation[22][23] . Decision trees classify instances (data points) by sorting them down the tree from the root node to some leaf node. This lead node gives the classification of the instance. Each branch of the decision tree represents a possible scenario of decision and its outcome.

Decision tree algorithms depict concept descriptions in the form of a tree structure. They begin learning with a set of instances and create a tree structure that is used to classify new instances. An instance in a dataset is described by a set of feature values called attributes, which can have either continuous or nominal values. Decision tree induction is best suitable for data where each example in the dataset is described by a fixed number of attributes for all examples of that dataset. Decision tree methods use a divide and conquer approach. They can be used to classify an example by starting at the root of the tree and moving through it until a leaf node is reached, which provides the classification of the instance.

Each node of a decision tree specifies a test of some attribute and each branch that descends from the node corresponds to a possible value for this attribute. The following example illustrates a simple decision tree.



3. TREPAN ALGORITHM

The TREPAN [24] and [25] algorithms developed by Craven et al are novel rule-extraction algorithms that mimic the behavior of a neural network. Given a trained Neural Network, TREPAN extracts decision trees that provide a close approximation to the function represented by the network. In this work, we are concerned with its application to trained variety of learned models as well. TREPAN uses a concept of recursive partitioning similar to other decision tree induction algorithms. In contrast to the depth-first growth used by other decision tree algorithms, TREPAN expands using the best first principle. Thus node which increases the fidelity of the fidelity of the tree when expanded is deemed the best.

In conventional decision tree induction algorithms the amount of training data decreases as one traverses down the tree by selecting splitting tests. Thus there is not enough data at the bottom of the tree to determine class labels and is hence poorly chosen. In contrast TREPAN uses an 'Oracle' to answer queries, in addition to the training samples during the inductive learning process. Since the target here is the function represented by the neural network, the network itself is used as the 'Oracle'. This learning from larger samples can prevent the lack of examples for the splitting tests at lower levels of the tree, which is usually a problem with conventional decision tree learning algorithms. It ensures that there is a minimum sample of instances available at a node before choosing a splitting test for that node where minimum sample is one of the user specified parameters. If the number of instances at the node, say m is less than minimum sample then TREPAN will make membership queries equal to (minimum sample m) from the 'Oracle' and then make a decision at the node. The following illustrates a pseudocode of the TREPAN algorithm [26].

Algorithm : TREPAN

Input: Trained neural network; training examples $\{X_i Y_i\}$ = where yi , is the class label predicted by the trained neural network on the training example Xi , global stopping criteria.

Output : extracted decision tree

Begin

Initialize the tree as a leaf node

While global stopping criteria are not met and the current tree can be further refined

Do

Pick the most promising leaf node to expand

Draw sample of examples

Use the trained network to label these examples

Select a splitting test for the node

For each possible outcome of the test make a new leaf node

End

End

3.1 M-of-N Splitting tests

TREPAN uses the m-of-n test to partition the part of the instance space covered by a particular internal node. An m-of-n expression (a Boolean expression) is fulfilled when at least an integer threshold m of its n literals hold true. For example, consider four features a, b, c and d; the m-of-n test: 3-of-{a, b > 3.3, c, d} at a node signifies that if any of the 3 conditions of the given set of 4 are satisfied then an example will pass through that node. TREPAN employs a beam search method with beam width as a user defined parameter to find the best m-of-n test. Beam search is heuristic best-first each algorithm that evaluates that first n node (where n is a fixed value called the 'beam width') at each tree depth and picks the best out of them for the split. TREPAN uses both local and global stopping criteria. The growth of the tree stops when any of the following criteria are met: the size of the tree which is a user specific parameter or when all the training examples at node fall in the same class.

3.2 Single Test TREPAN and Disjunctive TREPAN

In addition to TREPAN algorithm, Craven has also developed two of its important variations. The single test TREPAN algorithm is similar to TREPAN in all respects except that as its name suggests it uses single feature tests at the internal nodes. Disjunctive TREPAN on the other hand, uses disjunctive "OR" tests at the internal nodes of the tree instead of the m-of-n tests. A more detailed explanation of the TREPAN algorithm can be found in Craven's dissertation [27].

Baesens et al [28] have applied TREPAN to credit risk evaluation and reported that it yields very good classification accuracy as compared to the logistic regression classifier and the popular C4.5 algorithm.

4. C4.5 ALGORITHM

The C4.5 algorithm [29] is one of the most widely used decision tree learning algorithms. It is an advanced and incremental software extension of the basic ID3 algorithm [30] designed to address the issues that were not dealt with by ID3. The C4.5 algorithm has its origins in Hunt's Concept Learning Systems (CLS) [31]. It is a non-incremental algorithm, which means that it derives its classes from an initial set of training instances. The classes derived from these instances are expected to work for all future test instances. The algorithm uses the greedy search approach to select the best attribute and never looks back to reconsider earlier choices. The C4.5 algorithm searches through the attributes of the training instances and finds the attribute that best separates the data. If this attribute perfectly classifies the training set then it stops else it recursively works on the remaining in the subsets (m = the remaining possible values of the attribute) to get their best attribute. Some attributes split the data more purely than others. Their values correspond more consistently with instances that have particular values of the target class. Therefore it can be said that they contain more information than the other attributes. But there should be a method that helps quantify this information and compares different attributes in the data which will enable us to decide which attribute should be placed at the highest node in the tree.

4.1 Information Gain, Entropy Measure and Gain Ratio

A fundamental part of any algorithm that constructs a decision tree from a dataset is the method in which it selects attributes at each node of the tree for splitting so that the depth of the tree is the minimum. ID3 uses the concept of Information Gain which is based on Information theory [32] to select the best attributes. Gain measures how well a given attribute separates training examples into its target classes. The one with the highest information is selected. Information gain calculates the reduction in entropy (or gain information) that would result from splitting the data into subsets based on an attribute.

The information gain of example set S on attribute A is defined as,

$$Gain(S, A) = Entropy(S) - \sum \frac{|S_v|}{|S|} Entropy(S_v) \qquad \text{Eq.1}$$

In the above equation, S is the number of instances and |Sv| is a subset of instances of S where A takes the value v. Entropy is a measure of the amount of information in an attribute. The higher the entropy, the more the information is required to completely describe the data. Hence, when building the decision tree, the idea is to decrease the entropy of the dataset until we reach a subset that is pure (a leaf), that has zero entropy and represents instances that all belong to one class. Entropy is given by,

$$Entropy(S) = \sum - p(I) \log_2 p(I) \qquad \text{Eq.2}$$

where p(I) is the proportion of S belonging to Class I.

Suppose we are constructing a decision tree with ID3 that will enable us to decide if the weather is favorable to play football. The input data to ID3 is shown in table 2 below adapted from Quinlan's C4.5.

Table 2. Play Tennis Examples Dataset

Day	Outlook	Temperature	Humidity	Wind	Play Tennis
1	Sunny	Hot	High	Weak	No
2	Sunny	Hot	High	Strong	No
3	Overcast	Hot	High	Weak	Yes
4	Rain	Mild	High	Weak	Yes
5	Rain	Cool	Normal	Weak	Yes
6	Rain	Cool	Normal	Strong	No
7	Overcast	Cool	Normal	Strong	Yes
8	Sunny	Mild	High	Weak	No
9	Sunny	Cool	Normal	Weak	Yes
10	Rain	Mild	Normal	Weak	Yes
11	Sunny	Mild	Normal	Strong	Yes
12	Overcast	Mild	High	Strong	Yes
13	Overcast	Hot	Normal	Weak	Yes
14	Rain	Mild	High	Strong	No

In this example,

$$Entropy(S) = -\left(\frac{9}{14}\right)\log_2\left(\frac{9}{14}\right) - \left(\frac{5}{14}\right)\log_2\left(\frac{5}{14}\right) = 0.9450 \qquad \text{Eq.3}$$

(Note: Number of instances where play tennis = yes is 9 and play tennis = No is 5)

The best attribute of the four is selected by calculating the Information Gain for each attribute as follows,

$$Gain\,(S, Outlook\,) = Entr\,.(S) - \frac{5}{14}Entr\,.(Sunny\,) - \frac{4}{14}Entr\,.(Overcast\,) - \frac{5}{14}Entr\,.(Rain\,) \qquad \text{Eq.4}$$

$$Gain(S, Outlook) = 0.9450 - 0.3364 - 0 - 0.3364 = 0.2670$$

Similarly, $Gain(S, Temp) = -0.42$ and $Gain(S, Wind) = 0.1515$

The attribute outlook has the highest gain and hence it is used as the decision attribute in the root node. The root node has three branches since the attribute outlook has three possible values, (Sunny, Overcast, and Rain). Only the remaining attributes are tested at the sunny branch node since outlook has already been used at the node. This process is recursively repeated until: all the training instances have been classified or every attribute has been utilized in the decision tree. The ID3 has a strong bias in favor of tests with many outcomes. Consider an employee database that consists of an employee identification number. Every attribute intended to be unique and partitioning any set of training cases on the values of this attribute will lead to a large number of subsets, each containing only one case. Hence the C4.5 algorithm incorporates use of a statistic called the "Gain Ratio" that compensates for the number of attributes by normalizing with information encoded in the split itself.

$$GainRatio = \frac{Gain(S, A)}{I(A)} \quad \text{Eq.5}$$

In the above equation,

$$I(A) = \sum - p(I_A)\log_2 p(I_A) \quad \text{Eq.6}$$

C4.5 has another advantage over ID3; it can deal with numeric attributes, missing values and noisy data.

5. EXPERIMENTATION AND RESULT ANALYSIS

We analyze three datasets with classes greater than two and we compare the results of Single-test TREPAN and C4.5 with that of X-TREPAN in terms of comprehensibility and classification accuracy. A generalized feed forward network was trained in order to investigate the ability of X-TREPAN in comprehending GFF networks. The traditional 'using-network' command was used to validate that X-TREPAN was producing correct outputs for the network. In all the experiments, we adopted the Single-test TREPAN as the best variant for comparison with the new model.

5.1 Body Fat

Body Fat is a regression problem in the simple machine learning dataset category. The instances are sought to predict body fat percentage based on body characteristics. A 14-4-1 MLP with hyperbolic tangent function was used to train the network for 1500 epochs giving an r (correlation co-efficient) value of 0.9882. Figure 6 shows the comparison of classification accuracy of body fat by the three models.

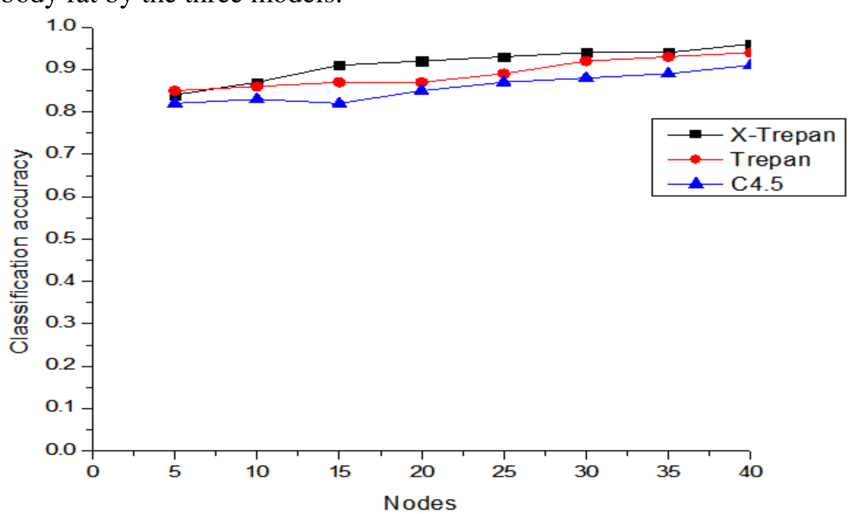


Figure 6. Comparison of classification accuracy of Body fat by the three algorithms

TREPAN achieves a classification accuracy of 94% and C4.5 produces a classification accuracy of 91% while X-TREPAN achieves a comparatively much higher accuracy of 96%. Additionally, both X-TREPAN and TREPAN generate similar trees in terms of size but accuracy and comprehensibility attained by X-TREPAN are comparatively higher.

The tables below show the confusion matrix of the classification accuracy achieved by TREPAN in comparison with X-TREPAN. While TREPAN produces a classification accuracy of 92.06% X-TREPAN produces a comparatively much higher accuracy of 96.83% as indicated in Table 3 below.

Table 3. Body Fat Confusion Matrix (X-TREPAN)

Actual/Predicted	Toned	Healthy	Flabby	Obese
Toned	13	0	0	0
Healthy	1	21	0	0
Flabby	0	0	9	1
Obese	0	0	0	18
Classification Accuracy (%)	92.86%	100.00%	100.00%	94.74%
Total Accuracy (%)		96.83%		

Table 4. Body Fat Confusion Matrix (TREPAN)

Actual/Predicted	Toned	Healthy	Flabby	Obese
Toned	13	0	0	0
Healthy	1	20	0	0
Flabby	0	0	9	0
Obese	0	0	3	16
Classification Accuracy (%)	92.86%	95.24%	75.00%	100.00%
Total Accuracy (%)		92.06%		

Additionally, both TREPAN and X-TREPAN generate identical trees in terms of size but accuracy attained by X-TREPAN is comparatively higher.

5.2 Outages

Outages constitute a database from the small dataset category. A12-3-1 MLP network with a hyperbolic tangent and bias axon transfer functions in the first and the second hidden layer respectively gave the best accuracy. The model was trained for 12000 epochs and achieved an r (correlation co-efficient) value of 0.985 (or an r2 of (0.985)2). Figure 7 shows the comparison of classification accuracy of outages by the three algorithms.

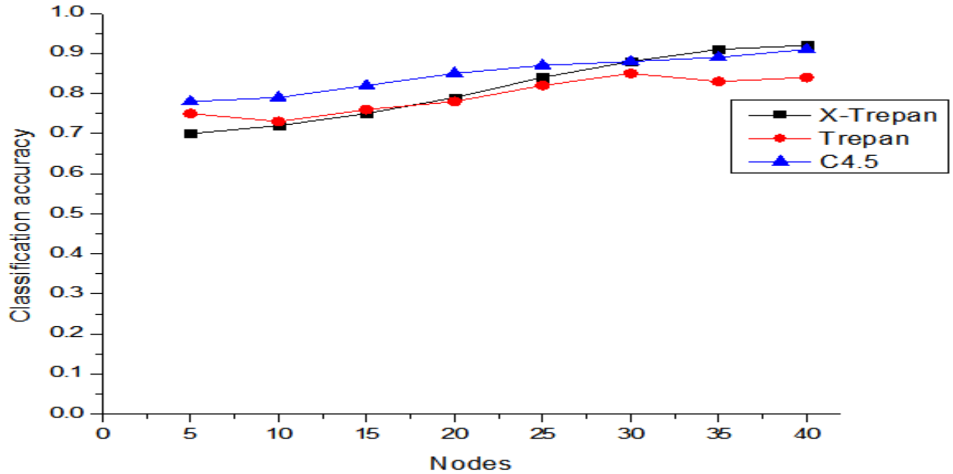


Figure 7. Comparison of classification accuracy of Outages by the three algorithms

In terms of classification accuracy, as can be seen in the figure above, TREPAN achieves 84%, C4.5 achieves 91% while X-TREPAN achieves 92%.

However, here TREPAN, C4.5 and X-TREPAN all generate very different trees in terms of size with C4.5 producing the largest and most complex decision tree while X-TREPAN produces the simplest and smallest decision tree with comparatively higher accuracy and comprehensibility.

The tables below show the confusion matrix of the classification accuracy achieved by both algorithms. X-TREPAN achieves 85% while TREPAN achieves 76%.

Table 5. Outages Confusion Matrix (X-TREPAN)

Actual/Predicted	**C11**	**C12**	**C13**	**C14**	**C15**
C11	3	0	0	0	0
C12	4	48	5	0	0
C13	0	1	8	0	0
C14	0	0	1	5	0
C15	0	0	0	0	0
Classification Accuracy (%)	42.86%	97.96%	57.14%	100%	0.00%
Total Accuracy (%)			85.33%		

Table 6. Outages Confusion Matrix (TREPAN)

Actual/Predicted	**C11**	**C12**	**C13**	**C14**	**C15**
C11	2	5	0	0	0
C12	3	43	3	0	0
C13	0	6	7	1	0
C14	0	0	0	5	0
C15	0	0	0	0	0
Classification Accuracy (%)	40.00%	79.63%	70.00%	83.33%	0.00%
Total Accuracy (%)			76.00%		

5.3 Admissions

A typical University admissions database model based on a 22-15-10-2 MLP network. Two hidden layers with the hyperbolic tangent transfer functions were used for modeling. The best model was obtained by the X-TREPAN with a minimum sample size of 1000, tree size of 50 and classification accuracy of 74%. Figure 8 gives the comparison of classification accuracy of Admissions by the three models.

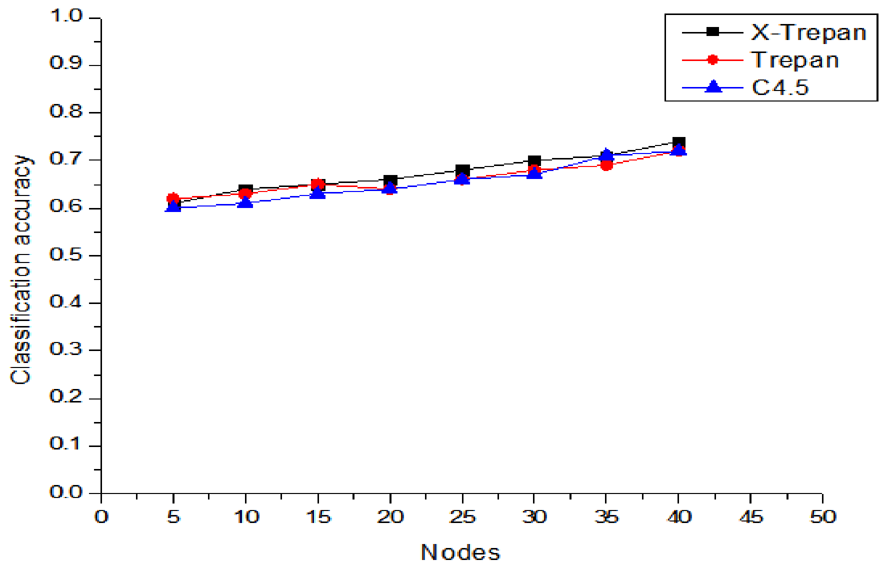


Figure 8. Comparison of classification accuracy of Admissions by the three algorithms

On the other hand, C4.5 achieved an accuracy of 71.97% (not rounded) almost equaling that of TREPAN of 72%, but produced a significantly large and complex decision tree.

In terms of Confusion Matrix, TREPAN achieved an accuracy of 71.6% very close to that of X-TREPAN. The confusion matrix is shown in the Tables below.

Table 7. Admissions Confusion Matrix (X-TREPAN)

Actural/Predicted	Yes	No
Yes	401	279
No	168	754
Classification Accuracy (%)	70.47%	72.99
Total Accuracy (%)	72.10%	

Table 8. Admissions Confusion Matrix (TREPAN)

Actural/Predicted	Yes	No
Yes	379	190
No	259	774
Classification Accuracy (%)	59.40%	80.29
Total Accuracy (%)	71.67%	

6. PERFORMANCE ASSESSMENT

6.1 Classification Accuracy

The classification accuracy or error rate is the percentage of correct predictions made by the model over a data set. It is assessed using the confusion matrix. A confusion matrix is a matrix plot of predicted versus actual classes with all of the correct classifications depicted along the diagonal of the matrix. It gives the number of correctly classified instances, incorrectly classified instances and overall classification accuracy.

The accuracy of the classifier is given by the formula,

$$Accuracy(\%) = \frac{(TP + TN)}{(TP + FN + FP + TN)} \times 100 \qquad \text{Eq.5}$$

Where true positive = (TP), true negative = (TN) false positive = (FP) and false negative = (FN). A false positive (FP) is when a negative instance incorrectly classified as a positive and false negative (FN) is when a positive instance is incorrectly classified as a negative. A true positive (TP) is when an instance is correctly classified as positive and true negative (TN) is when an instance is correctly classified as negative and so on.

A confusion matrix is a primary tool in visualizing the performance of a classifier. However it does not take into account the fact that some misclassifications are worse than others. To overcome this problem we use a measure called the Kappa Statistic which considers the fact that correct values in a confusion matrix are due to chance agreement.

The Kappa statistic is defined as,

$$\hat{k} = \frac{P((A) - P(E))}{1 - P(E)} \qquad \text{Eq.6}$$

In this equation, P(A) is the proportion of times the model values were equal to the actual value and, P(E) is the expected proportion by Chance.

For perfect agreement, Kappa = 1. For example: a Kappa statistic of 0.84 would imply that the classification process was avoiding 84% of the errors that a completely random classification would generate.

6.2 Comprehensibility

The comprehensibility of the tree structure decreases with the increase in the size and complexity. The principle of Occam's Razors says "when you have two competing theories which make exactly the same projections, the one that is simpler is the better" [33]. Therefore, among the three algorithms, X-TREPAN is better as it produces smaller and simpler trees as against Single-test TREPAN and C4.5 in most scenarios.

7. CONCLUSION

The TREPAN algorithm code was modified (X-TREPAN) to be able to work with multi-class regression type problems. Various experiments were run to investigate its compatibility with generalized feed forward networks. The weights and network file were restructured to present GFF networks in a format recognized by X-TREPAN. Neural Network models were trained on each dataset varying parameters like network architecture and transfer functions. The weights and biases obtained from the trained models of the three datasets were fed to X-TREPAN for decision tree learning from neural networks. For performance assessment, classification accuracy of Single-test TREPAN, C4.5 and X-TREPAN were compared. In the scenarios discussed in the paper, X-TREPAN model significantly outperformed the Single-test TREPAN and C4.5 algorithms in terms of classification accuracy as well as size, complexity and comprehensibility of decision trees. To validate the results, we use classification accuracy not as the only measure of performance, but also the kappa statistics. The kappa statistical values further validate the conclusions that X-TREPAN is a better one in terms of decision tree induction.

REFERENCES

[1] A. B. Ticle, R. Andrews, M. Golea, J. Diederich The truth will come to light: Directions and challenges in extracting the knowledge embedded within trained artificial neural networks, IEEE Trans. Neural Networks, vol. 9, no 6, pp. 1057-1068. 1998.

[2] G. Papantonopoulos, K. Takahashi, T. Bountis, B. G. Loos, Artificial Neural Networks for the Diagnosis of Aggressive Periodontitis Trained by Immunologic Parameters. PLoS ONE 9(3): e89757. (doi:10.1371/journal.pone.0089757), 2014.

[3] D. Puzenat, Behavior Analysis through Games Using Artificial Neural Networks, Third International Conference on Advances in Computer-Human Interactions (IEEE), pp. 134–138, Saint Maarten, France, 2010.

[4] M. Mano, G. Capi, N. Tanaka, S. Kawahara, An Artificial Neural Network Based Robot Controller that Uses Rat's Brain Signals, Robotics, 2,54-65, doi:10.3390/robotics2020054. (www.mdpi.com/journal/robotics), 2013.

[5] E. V. Raghavendra, P. Vijayaditya, K. Prahallad, speech generation National Conference on Communications (NCC), Hyderabad, India 2010. (doi: 10.1109/NCC.2010.5430190)

[6] A. Hossain, M. Rahman, U. K.Prodhan, F. Khan, Implementation Of Back-Propagation Neural Network For Isolated Bangla Speech Recognition, International Journal of Information Sciences and Techniques (IJIST) vol. 3, no. 4, 2013.

[7] S. Ayat, Z. A. Pour, Comparison between Artificial Neural Network Learning Algorithms for Prediction of Student Average considering Effective Factors in Learning and Educational Progress, Journal of mathematics and computer science 8, pp. 215 – 225, 2014.

[8] R. Andrews, J. Diedrich, A. Ticle, Survey and critique of techniques for extracting rules from trained artificial neural networks, Knowledge-Based System, vol. 8, no 6, pp. 373-389. 1995.

[9] A. B. Ticke, R. Andrews, M. Golea, J. Diederich, The truth will come to light: Directions and challenges in extracting the knowledge embedded within trained artificial neural networks, IEEE Trans. Neural Network, vol. 9 no. 6, pp. 1057 - 1068, 1998.

[10] A. Ticke, F. Maire, G. Bologna, R. Andrews, J. Diederich, Lessons from past current issues and future research directions in extracting knowledge embedded in artificial neural networks in Hybrid Neural Systems, New York: Springer-Verlag, 2000.

[11] H. Lohminger, Teach/Me Data Analysis, Springer-Verlag, Berlin-New York-Tokyo, 1999.

[12] C. Aldrich, Exploratory analysis of metallurgical process data with neural networks and related methods, Elsevier 2002.

[13] StatSoft, Inc. (2004). Electronic Statistics Textbook. Tulsa, OK: StatSoft.

[14] M. M. Nelson, W.W. Illingworth, A Practical Guide to Nerual Nets. 4th ed. 1992.

[15] K. Plunkeet, J. L. Elman, (1997). Exercises in Rethinking Innateness: A handbook for connectionist simulations, MIT Press, pp. 166.

[16] G.G. Towell, J. W. Shavli, Extracting Refined Rules from Knowledge-based Neural Networks, Machine Learning, 13:71-101 1993.

[17] M.W Craven, J. W. Shavlik, Using sampling and queries to extract rules from trained neural networks, Machine Learning. Proceedings of the Eleventh International Conference, Cohen W. W & Hirsh H.(Eds), San Francisco, CA:Morgan Kaufmann 1995.

[18] L. Fu, Rule Learning by searching on adopted nets, In Proceedings of the 9th National Conference on Artificial Intelligence, Anaheim, CA, pp.590-595, 1991.

[19] S. Thrums, Extracting rules from artificial neural networks with distributed representations, In Advances in Neural Information Processing Systems 7, Cambridge, MA:Mit press, pp 505-512, 1995.

[20] M. W. Craven, Extracting Comprehensible models from trained Neural Network, PhD Thesis, Computer Science Department , University of Wisconsin, Madison, WI, 1996.

[21] R. Andrews, J. Diederich, A. B. Tickle, A survey and critique of techniques for extracting rules from trained neural networks. Knowledge Based Systems, 8(6), 373-389, 1995.

[22] D. Biggs, B. de Ville, E. Suen, A method of choosing multiway partitions for classification and decision tree. Journal of Applied Statistics 18(1): 49-62, 1991.

[23] G. Liepins, R. Goeltz, R. Rush, Machine Learning techniques for natural resource data analysis . AI Applications 4(3):9-18, 1990.

[24] M. W. Craven, J. W. Shavlik, Using sampling and queries to extract rules from trained neural networks, Machine Learning. Proceedings of the Eleventh Inter-national Conference, Cohen W. W & Hirsh H. (Eds), San Francisco, CA. 1995.

[25] M. W. Craven, J. W. Shavlik, Extracting tree-structured representations of trained networks. In Advance in Neural Information Procession Systems, volume 8, pages 24-30, Denver, Com, MIT Press. 1996.

[26] F. Chen, Learning accurate and understandable rules from SVM classifiers, M. Sc. Thesis, School of computing science, Simon Fraser University. 2004.

[27] M.W. Craven, Extracting Comprehensible models from trained Neural Network, PhD Thesis, Computer Science Department, University of Wisconsin, Madison, WI. 1996.

[28] B. Baesens, R. Setiono, C. Mues, J. Vanthienen, Using Neural Network Rule Extraction and Decision Tables for Credit-Risk Evaluation management Science, vol. 49, no. 3, pp. 312, 2003.

[29] J. R. Quinlan, C4.5: Programs for Machine Learning. San Mateo, CA: Morgan Kaufmann, 1993.

[30] J. Ross Quinlan, ID3 Algorithm, Machine Learning, University of Sydney, book vol. 1, no.1, Syney, Australia. 1975.

[31] E. Hunt, J. Marin, P. Stone, Experiments in Induction, New York, Academic Press, 1966.

[32] L. J. Breiman, R. Friedman, C. Olshen, Classification and regression trees, Wadsworth and Brooks, Belmont, CA. 1984.

[33] L. Breiman, J. Friedman, R. Olshen, C. Stone, Classification and regression trees, Wadsworth and Brooks, Belmont, CA. 1984.

THE DETERMINANTS OF CLOUD COMPUTING ADOPTION IN SAUDI ARABIA

Abdullah Alhammadi, Clare Stanier, Alan Eardley

Faculty of Computing, Engineering and Sciences, Staffordshire University,
Beaconside, Stafford, Staffordshire, United Kingdom ST18 0AD
a.alhammadi@staffs.ac.uk, c.stanier@staffs.ac.uk,
w.a.eardley@staffs.ac.uk

ABSTRACT

There is a large volume of published studies investigating the factors that affect cloud adoption. However, there are very few studies which investigate cloud computing adoption in technologically developing countries and one focus of the research was to examine whether the factors which influence cloud computing adoption in technologically developed countries also apply in technologically developing countries. The research presented in this paper builds on the diffusion of innovation theory (DOI) and the Technology-organisation-environment (TOE) framework in order to investigate the factors which influence cloud adoption. Fourteen hypothesis were developed from the literature on cloud adoption and were examined in the research. Data was collected by using a web-based questionnaire and was analysed using a range of statistical measures. This paper discusses the design and implementation of the study, the data analysis and conclusions from the analysis and compares the findings of this study with the findings of similar studies in technologically developed countries. The study shows that there are some similarities as well as some differences in the factors that affect cloud adoption between technologically developed countries and technologically developing countries.

KEYWORDS

Cloud computing adoption, Diffusion of innovation (DOI), Technology-organisation-environment (TOE)

1. INTRODUCTION

Cloud computing has been described as the next generation model of computing [1]. The cloud computing was defined by National Institute of Standards and Technology NIST as a model for enabling convenient, on-demand network access to a shared pool of configurable computing resources […] that can be rapidly provisioned and released with minimal management effort or service provider interaction" (NIST, 2011). Cloud computing has four deployment models namely public cloud, private cloud, hybrid cloud and community cloud. Enterprise spending on cloud computing is increasing at five times the rate of traditional IT systems (cited by [2]. However, cloud computing adoption rates vary between technologically developed countries and technologically developing countries [3]. The literature on Cloud Computing adoption focuses primarily on adoption in technologically developed countries. The literature does not include any empirical study carried out to investigate cloud adoption in the Kingdom of Saudi Arabia (KSA)

David C. Wyld et al. (Eds) : CSEN, AISO, NCWC, SIPR - 2015
pp. 55–67, 2015. © CS & IT-CSCP 2015 DOI : 10.5121/csit.2015.51406

at the organisational level. There is only one study which investigates cloud adoption in KSA [4] and this focused on the user acceptance of cloud computing at an individual level while factors related to an organisational level were ignored. This study discussed the factors that affect cloud computing decision by integrating DOI and TOE framework.

2. THEORETICAL BACKGROUND

The conceptual model that is developed in this paper is based on concepts derived from the diffusion of innovation (DOI) theory and Technology, Organisation, Environment (TOE) framework. DOI, first proposed by Rogers (1962), is concerned with the way in which new ideas are adopted within organisations over time and how ideas influence change within organisations. DOI identifies five factors that affect the adoption of new ideas: relative advantage, compatibility, complexity, trialability and observability [5]. TOE framework was developed by Tornatzky and Fleischer in 1990 to investigate innovation adoption at organisation level [6], [7] and the framework provides a holistic picture of the factors that influence the adoption of technology [7], [8]. TOE categorises the factor which influence technology adoption in organisation into three categories namely, technological context, organisation context, and environmental context [8]. It has been argued that the use of these three elements gives the TOE framework the advantage over other technology adoption theories in studying technology use, adoption and the value added from technology innovation [7]. Numerous studies have attempted to investigate the adoption of new technologies from both individual and organizational perspective [9]–[11]. These studies have tried to identify factors which influence technology adoption and the relationship between these factors and technology adoption [7]. Technology adoption theory plays an important role in investigation the adoption of technology in different disciplines such as E-commerce, RFID, ERP [12]–[16]. Several studies have used outsourcing theories to investigate cloud adoption [17] . Technology adoption theories have been used to support the investigation of cloud computing adoption in a number of studies [10, 18, 19]. In this paper, we use a DOI/TOE approach as the focus of the study is on the adoption of cloud computing at organisational level.

From the literature on cloud computing adoption, we identified 14 factors which influence organisational decisions about the adoption of cloud computing. From these factors, we developed hypothesis as to the way in which the factors affect the adoption of cloud computing. Following TOE, we group these hypothesis into the TOE categories of technology context, organisational context and environment context and we also use a fourth category, the DOI category, which address elements identified though analysis based on DOI. The second phase of the research evaluated the hypothesis through use of a questionnaire. This section describes the categories used and way in which hypothesis were developed from the factors identified in the literature.

2.1. Technology context

The Technology context refers to a “comprised of the variables that influence an individual, an organization, and an industry’s adoption of innovation” [7]

2.1.1 Technology readiness

Cloud computing is a relatively new model of IT service delivery model. Thus, the technology context is a very important determinant to investigate in relation to the adoption of cloud computing. Technology readiness is described as the IT infrastructure available to an enterprise to obtain cloud services and human resources that can manage cloud services [10]. We therefore identify as hypothesis 1:

H1: technology readiness positively influences cloud computing adoption.

2.1.2 Security

Security is seen as one of the highest risk elements in the adoption of cloud computing [18]–[20] although it has also been argued that there is no relationship between security and cloud adoption [10]. This inconsistency is tested by hypothesis 2:.

H2: security concerns negatively influence cloud computing adoption.

2.1.3 Technical barrier

Technical issues such as the complexity of existing IT systems, portability and interoperability and vendor lock-in have been identified as possible barriers to the adoption of cloud computing [21].

H3: technology barrier negatively influence cloud computing adoption.

2.2. Organisational context

The organisational context refers to the characteristics of the organisation and its internal resources [9], [22]. Organisation characteristics include organisation size, status, industry, and scope. While the internal resources include knowledge capability, top management support and organisation readiness [7]. Cloud computing has technological as well as organisational implication, this makes the organisational context a key determinant of cloud adoption.

2.2.1 Enterprise size

Enterprise size is considered to be one of the main factors influencing technical innovation [9], [23]–[25]. It has been argued that large enterprises are more likely to adopt newer technologies such as ERP and e-commerce [24], [26]. The main reasons given being that larger enterprise have greater organisational and financial resources. However, the financial model of cloud computing services makes it attractive to SMEs meaning that enterprise size is one of the determinants to investigate in relation to cloud computing adoption.

H4: enterprise size has a positive influence to adopt cloud computing services.

2.2.2 Top management support

Top management support refers to the decision makers who influence the adoption of innovation [27] and is which described by [28] as goal specificity, resource management, and commitment. Thus, the role of top management support is crucial to success the adopting of new technology. Therefore, Top management support is regarded in many studies as an important factor influencing the adoption of technology innovation [29]–[31].

H5: top management support has a positive impact on cloud adoption.

2.2.3 Organisation readiness

Organisation readiness aims to measure if the organisation has the capability to adopt the innovation [6] and is defined as the availability of human, technology and financial resources to adopt cloud computing [6], [13], [32]. Knowledge about cloud computing and the attitudes towards using the technology are an important factors in the adoption of cloud computing.

H6: organisation readiness has a positive impact on cloud adoption.

2.2.4 Enterprise status

Enterprise status is defined in this study as having either status as an establish company or as a Start-up company. There is no empirical study in the literature which investigates the impact of enterprise status impact on technology adoption in general and cloud adoption specifically. However, some studies have linked enterprise status and cloud computing adoption [33], [34].

H7: enterprise status has a positive impact on cloud adoption.

2.2.5 Industry sector

IT has a major impact on how enterprises manages their business but the role of IT in an organisation differs, depending on the sector [35]. Moreover, technology adoption varies between sectors, for example, the largest user of technology is the financial sector [25]. However, sensitivity of financial data means that the financial sector may be more cautious when adopting cloud computing [36], [37].

H8: industry sector has a positive impact on cloud adoption.

2.3. Environmental context

Environmental context refers to the external factors that influence the adoption of technology includes government regulation and initiative, service providers and competitors [7]. This study will investigate the following factors in terms of environmental context; competitive pressure, external support and government support.

2.3.1 Competitive pressure

Competitive pressure may be a factor which influences the adoption of cloud computing although there is no consensus on this issue. It has been argued that competitive pressure is very influential in the adoption of technology in general [6] although one study found that there is no relation between the competitive pressure and the adoption of cloud computing technology [19].

H9: competitive pressure has a positive impact on cloud adoption.

2.3.2 External support

External support in this research is defined as support from the cloud service provider to influence clients to adopt cloud technology. There is a lack of understanding about cloud services issues such as cloud architecture and pricing models [38]. This could represent one of the barriers for SMEs to adopt cloud computing. CSPs can provide knowledge and expertise to their clients [39].

H10: External support has a positive impact on cloud adoption.

2.3.3 Government support

Government support in this context is understood as the regulation, policies and initiatives that support enterprises in the adoption of cloud computing. Government regulation can play an important role in the adoption of technology innovation [9], [26]. Regulation can encourage or discourage cloud computing adoption [9], [22]. One study found that government regulation had more influence on the adoption of E-business in developing countries comparing with developed countries [27]. A study in KSA found that Saudi SMEs seek the support from the government of Saudi Arabia in relation to technology adoption [43].

H11: Government support has a positive impact on cloud adoption.

2.4. Diffusion of Innovation Category

2.4.1 Relative advantage

Relative advantage is defined as "the degree to which an innovation is perceived as being better than the idea it superseded" [40]. Cloud computing offers technical as well as economic advantages over traditional IT environments.

H12: relative advantage has a positive impact on cloud adoption.

2.4.2 Compatibility

Compatibility refers to which extent the new innovation fits with existing organisation's values, culture and practices [9], [40]. From a technical perspective, the extent to which cloud solutions are compatible with existing systems is a key factor when considering moving to a cloud environment. In addition, the extent to which cloud computing services are compatible with statutory regulation is crucial for organisation when considering moving to cloud computing.

H13: compatibility has a negative impact on cloud adoption.

2.4.3 Complexity

Complexity in DOI terms was defined as “the degree to which an innovation is perceived as relatively difficult to understand and use” [44]. Cloud computing comes with some challenges including security and privacy and the use of advanced technology; adopting cloud computing requires new skills and expertise to manage cloud solutions [9]. Thus, these issues will affect cloud adoption.

H14: complexity has a negative impact on cloud adoption.

3. RESEARCH METHOD

In order to test the hypothesis in a technically developing country, a web questionnaire was used with computing professionals in KSA. The questionnaire was administered in English and in Arabic. The sample frame refers to the set of people/enterprises from the targeted population that have the change to be selected [41]. In order to select the respondents, two approaches were used to select the sample. Two e-services provider in KSA agreed to distribute questionnaire to their clients. The CSPs used included ELM which is one of the leading CSPs in KSA. To avoid limiting the sample to respondents from CSPs, the questionnaire was also distributed using a professional network. LinkedIn is a professional social network that brings together users with the same interests.

Piloting the questionnaire is a process to test the questionnaire reliability, validity and error testing [42]. To enhance the validity of the questionnaire, a pilot study was conducted in two phases. The first stage of the pilot was conducted to test the understandability of the questionnaire and to ensure the different language versions had the same meaning. In addition, to what extent the questionnaire is readable with different devices (laptop, IPad and Mobile) was examined in this stage. The questionnaire was piloted with eight IS professionals from industry and academia who speak Arabic and English fluently. The second phase of the pilot study was conducted with the CSP. The ELM marketing team who have experience in questionnaire design and analysis evaluated the questionnaire. Some changes were made following the feedback from ELM, including changing some phrasing in Arabic to ensure understandability.

Following the pilot stage, a web based questionnaire platform was used. The questionnaire was built by using Survey Monkey with two version one in Arabic and one in English. The link was circulated by the CSPs to their clients and made available via LinkedIn. At the questionnaire deadline, 103 questionnaires had been completed. Following data cleansing, 81 valid questionnaires were selected for analysis, representing 81 companies. Reasons for excluding questionnaires from analysis included errors in completion which undermined the validity of the response and too man ‘don’t know’ or blank answers for analysis.

3.1. Data analysis

The first set of questions were designed to identify the characteristics of the enterprises which took part in the survey as this study aims to investigate cloud computing adoption from organisation level. The graph below shows the number of enterprises that participated in this survey in terms of the industry sectors that belong to it.

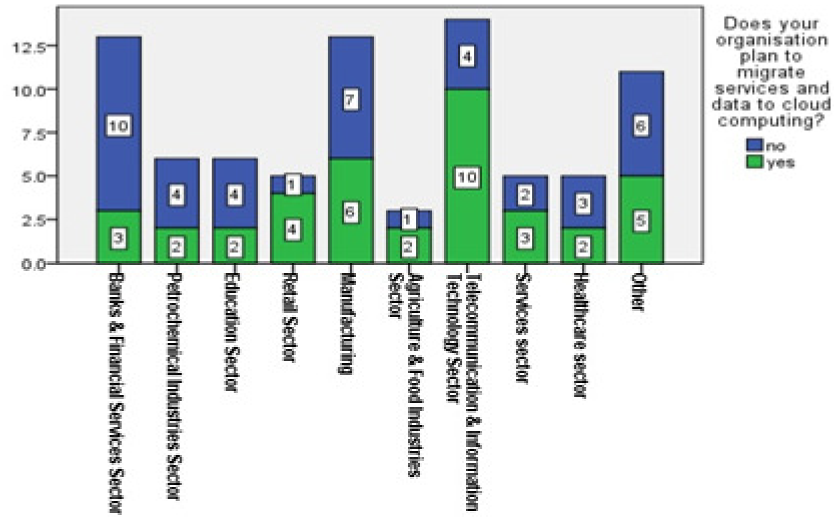


Figure 1: The number of cloud adoption among the industry

In addition, participants were asked to indicate whether their enterprises work on private or government sectors. More than half of respondents belong to the government sectors while about 47% of participants belong to private sector.

Table 1 illustrates the distribution of enterprises according to its size and status. In addition, the table explains the percentage of enterprise that adopted or plan to adopt cloud computing services.

Table 1. profile of enterprises

Variable	**Group**	**Frequency**	**Percentage**	**Cloud adopter**
Size	Small	18	22.2%	55.6%
	Medium	13	16%	69.2%
	Large	50	61.7%	40%
Sector	Government	43	53.1%	48.7%
	Private e	38	46.9%	51.3%
Enterprise status	Start-up	20	24.7%	85%
	Established	55	67.9%	40.1%
	Don't know	6	7.2%	0.0%

3.2. Inferential Analysis

Logistic regression was used to test the hypothesis described in section. The independent variables are the fourteen hypothesis where is the dependent variable is cloud computing adoption, a binary values were used which is 1 if the firm adopt or plan to adopt cloud computing and 0 otherwise. The independent variables are shown on the table below.

Table 2. Logistic Regression Results

	B	S.E.	Wald	df	Sig.	Exp(B)
Technology readiness	.391	.780	.252	1	.616	1.479
Security concerns	-1.147	.435	6.951	1	.008	.318
Technology Barriers	.459	.397	1.336	1	.248	1.583
Organisational readiness	1.989	.946	4.425	1	.035	7.312
Firm Size	-.900	.660	1.856	1	.173	.407
Firm Status	-2.936	1.218	5.811	1	.016	.053
Industry Sector	.619	.888	.485	1	.486	1.856
›p Management Support	1.768	.546	10.476	1	.001	5.858
Competitive pressure	-.176	.935	.035	1	.851	.839
External support	.396	.920	.185	1	.667	1.486
Government support	-1.774	.699	6.453	1	.011	.170
Relative advantage	.606	.900	.454	1	.501	1.833
Compatibility	1.039	.507	4.202	1	.040	2.827
Complexity ---------	-1.080	.751	2.069	1	.150	.340
Constant	-3.611	3.852	.878	1	.349	.027

After coding the hypothesis and calculate the mean of the items for each hypothesis, the binary logistic regression run. The independent variable is significant when the P value is less than 0.05 [43]. Consequently, it can be seen from the data in Table # that there are six predictors were found statistically significant which are security concerns, organisational readiness, firm status, top management support, government support and compatibility. Three of them belong to the organisational readiness group**.**

4. DISCUSSION

4.1. Technological context

Among the three dimensions of technological context, in this research, security was found to have a statistically significant relationship with cloud adoption. Thus, security concerns have a negative impact of adopting cloud computing. This is consistent with the findings in the earlier literature on cloud computing adoption although a 2014 study found that security was not a significant factor in cloud computing adoption; that study suggested that this was due to improvements in security compared to the earlier phase of cloud computing [10]. Unexpectedly, in this study, technology readiness and technology barriers were not found statistically significant. This result are consistent with a 2011 study in a technologically developing country which found that technology readiness were not significant predictor [48] although the 2011 findings were different to those of a study conducted in a technically developed country[10]. One possible explanation for this might be that there is a difference between attitudes of decision makers in technologically developed countries and technologically developing countries.

4.2 Organisational context

Interestingly, in this investigation, three predictors, organisation readiness, top management support and enterprise status, were found to significantly influence the decision about cloud computing adoption. In the literature, only one study was found which identified organisation readiness as a factor that influences cloud computing adoption [20]; that study identified a significant relationship between organisation readiness and cloud adoption which is similar to the findings of this study. In terms of enterprise status, based on the literature review there is no prior study examine firm status empirically in relation to cloud adoption or technology adoption in general. In this study, start-up enterprises were found to be more likely to adopt cloud computing. This may reflect the availability of infrastructure within KSA as established companies will already have developed infrastructure. Top management support was found to be a more significant factor influencing cloud adopting with P value at 0.001. The finding on top management support is in line with the findings of previous studies [9], [20], [44]–[46]. Contrary to expectations, this research did not find a significant relationship between the enterprise size and industry sectors. This is in contrast to earlier findings [9], [45], [46] which found a significant relationship between cloud adoption and enterprise size.

4.3. Environmental context

The environmental context was examined from three dimensions namely competitive pressure, external support and government support. The only factor found to be statistically significant is the government support. This finding is different from that of [9], [44]. These differences can be explained by the fact that Governments in technological developing counties play a major role in supporting enterprises in adopting new technology in terms of regulation and initiatives [47], [48].

One unanticipated finding was that external support did not significantly affect cloud adoption. This differs from the results of [19]who found computer supplier support has a significant effect on cloud adoption. However, the study conducted by [19],was carried out in a technically developed country. In the research presented here, competitive pressure was not found to a significant factor influencing the adoption of cloud technology. . This result is in agreement with those obtained by [9], [45]. In contrast, [23] and [48] found that the competitive pressure has a significant impact in association with cloud computing. A possible explanation for this is that the studies that found the competitive pressure has a significant factor were limited to one industry sector like [46] or limited to a small number of industry sectors like [20]. Table 3 summarises the findings of this research in respect of the hypothesis developed through the TOE and DOI.

4.4. Diffusion of Innovation Category

Unexpectedly, relative advantage was not found to be a significant factor in this study. This contrasts with findings from previous studies [9], [20], [44]–[46] which identified a significant relationship between relative advantage and cloud computing. An important finding in this research was that compatibility has a significant impact on cloud computing adoption. This finding is in accord with recent studies which indicated that the compatibility has a positive impact on cloud adoption [20], [45] although some studies report different findings [9], [44], [46]. Finally, it is somewhat surprising that in this study complexity was not found to significantly affect cloud adoption. This result is supported by other studies [44], [46] but differs from that of recent studies that found complexity negatively influences cloud computing adoption [9], [20], [45]. It should be noted that the respondents in this study had technical backgrounds.

Table 3. Summary of results

Technological context		
H1	Technology readiness	Rejected
H2	Security concerns	Accepted
H3	Technology Barriers	Rejected
Organisational context		
H4	Organisational readiness	Accepted
H5	Firm Size	Rejected
H6	Firm Status	Accepted
H7	Industry Sector	Rejected
H8	Top Management Support	Accepted
Environmental context		
H9	Competitive pressure	Rejected
H10	External support	Rejected
H11	Government support	Accepted
Diffusion of innovation		
H12	Relative advantage	Rejected
H13	Compatibility	Accepted
H14	Complexity	Rejected

5. CONCLUSION

This study integrates the DOI and TOE framework to investigate the factors that affect cloud computing adoption decision in Saudi Arabia. From the fourteen hypothesis were developed, there are only six hypothesis were found statistically significant. These factors are security concerns, organisation readiness, top management support, firm status, government support and compatibility. When evaluated against the literature on cloud computing adoption, it appears that security concerns and government support are more influential factors in technologically developing countries than in technologically developed countries. We found that start up companies were more likely to adopt cloud computing solutions. This may reflect the pricing model of cloud computing but may also reflect infrastructure differences between technically developed and technically developing countries. In addition,

The findings of this study contributes to both academia and industry. Cloud service providers can use the findings of this study to support marketing and decision making. The investigation shows that government has a major role to play in encouraging enterprises to adopt technology in general and specifically cloud computing in technologically developing countries. This study adds the body of knowledge about cloud computing adoption specifically in technologically developing countries.

ACKNOWLEDGEMENT

The author would like to thank ELM for distributing the questionnaire and piloting the questionnaire.

REFERENCES

[1] S. Rajan and A. Jairath, "Cloud Computing: The Fifth Generation of Computing," in 2011 International Conference on Communication Systems and Network Technologies, 2011, pp. 665–667.

[2] K. Ruan, J. Carthy, and T. Kechadi, "Cloud forensics: An overview," … Digit. forensics VII, vol. 0, 2011.

[3] E. O. Yeboah-Boateng and K. A. Essandoh, "Factors Influencing the Adoption of Cloud Computing by Small and Medium Enterprises in Developing Economies," Int. J. Emerg. Sci. Eng., no. 4, pp. 13–20, 2014.

[4] S. T. Alharbi, "Users' Acceptance of Cloud Computing in Saudi Arabia," Int. J. Cloud Appl. Comput., vol. 2, no. 2, pp. 1–11, 2012.

[5] F. C. Cua, "Information Systems Theory," in Information Systems Theory: Explaining and Predicting Our Digital Society, vol. 28, Y. K. Dwivedi, M. R. Wade, and S. L. Schneberger, Eds. New York, NY: Springer New York, 2012, pp. 303–333.

[6] B. Ramdani, D. Chevers, and D. a. Williams, "SMEs' adoption of enterprise applications: A technology-organisation-environment model," J. Small Bus. Enterp. Dev., vol. 20, no. 4, pp. 735–753, 2013.

[7] H. Gangwar, H. Date, and A. D. Raoot, "Review on IT adoption: insights from recent technologies," J. Enterp. Inf. Manag., vol. 27, no. 4, pp. 488–502, Jul. 2014.

[8] M. Nkhoma and D. Dang, "Contributing factors of cloud computing adoption: a technology-organisation-environment framework approach," Int. J. Inf. …, vol. 1, no. 1, pp. 38–49, 2013.

[9] T. Oliveira, M. Thomas, and M. Espadanal, "Assessing the determinants of cloud computing adoption: An analysis of the manufacturing and services sectors," Inf. Manag., vol. 51, no. 5, pp. 497–510, Jul. 2014.

[10] A. Rahimli, "Factors influencing organization adoption decision on cloud computing," Int. J. Cloud Comput. Serv. Sci., vol. 2, no. 2, pp. 140–146, 2013.

[11] S. S. Bharadwaj and P. Lal, "Exploring the impact of Cloud Computing adoption on organizational flexibility: A client perspective," in 2012 International Conference on Cloud Computing Technologies, Applications and Management (ICCCTAM), 2012, pp. 121–131.

[12] R. Ramanathan, U. Ramanathan, and L. W. L. Ko, "Adoption of RFID technologies in UK logistics: Moderating roles of size, barcode experience and government support," Expert Syst. Appl., vol. 41, no. 1, pp. 230–236, Jan. 2014.

[13] H.-F. Lin and S.-M. Lin, "Determinants of e-business diffusion: A test of the technology diffusion perspective," Technovation, vol. 28, no. 3, pp. 135–145, Mar. 2008.

[14] J. Tan, K. Tyler, and A. Manica, "Business-to-business adoption of eCommerce in China," Inf. Manag., vol. 44, no. 3, pp. 332–351, Apr. 2007.

[15] G. Feuerlicht and N. Margaris, "Cloud Computing Adoption : A comparative study," pp. 441–448, 2007.

[16] P. Gottschalk and H. Solli-Sæther, "Critical success factors from IT outsourcing theories: an empirical study," Ind. Manag. Data Syst., vol. 105, no. 6, pp. 685–702, 2005.

[17] I. Gonzenbach, C. Russ, and J. vom Brocke, "Make or Buy? Factors that Impact the Adoption of Cloud Computing on the Content Level," in Enterprise Content Management in Information Systems Research, J. vom Brocke and A. Simons, Eds. Berlin, Heidelberg: Springer Berlin Heidelberg, 2014, pp. 145–161.

[18] M. Carroll, A. van der Merwe, and P. Kotze, "Secure cloud computing: Benefits, risks and controls," in 2011 Information Security for South Africa, 2011, pp. 1–9.

[19] G. Chao, A. Peng, A. Dutta, and A. Choudhary, "Exploring Critical Risks Associated with Enterprise Cloud Computing," in CloudComp, vol. 133, V. C. M. Leung and M. Chen, Eds. Cham: Springer International Publishing, 2014, pp. 132–141.

[20] H. Gangwar, H. Date, and R. Ramaswamy, "Understanding determinants of cloud computing adoption using an integrated TAM-TOE model," J. Enterp. Inf. Manag., vol. 28, no. 1, pp. 107–130, Feb. 2015.

[21] N. Phaphoom, X. Wang, S. Samuel, S. Helmer, and P. Abrahamsson, "A Survey Study on Major Technical Barriers Affecting the Decision to Adopt Cloud Services," J. Syst. Softw., vol. 103, pp. 167–181, Feb. 2015.
[22] J. Baker, "The Technology–Organization–Environment Framework," in Information Systems Theory: Explaining and Predicting Our Digital Society, vol. 28, Y. K. Dwivedi, M. R. Wade, and S. L. Schneberger, Eds. New York, NY: Springer New York, 2012, pp. 231–245.
[23] M. G. Aboelmaged, "Predicting e-readiness at firm-level: An analysis of technological, organizational and environmental (TOE) effects on e-maintenance readiness in manufacturing firms," Int. J. Inf. Manage., vol. 34, no. 5, pp. 639–651, Oct. 2014.
[24] W.-Y. Pan, Ming-Ju and Jang, "DETERMINANTS OF THE ADOPTION OF ENTERPRISE RESOURCE PLANNING WITHIN THE TECHNOLOGY ORGANIZATION ENVIRONMENT FRAMEWORK : TAIWAN ' S COMMUNICATIONS INDUSTRY," J. Comput. Inf. Syst., vol. 48, no. 3, 2008.
[25] K. Zhu, K. Kraemer, and J. Dedrick, "Information technology payoff in e-business environments: an international perspective on value creation of e-business in the financial services industry," J. Manag. Inf. Syst., vol. 21, no. 1, pp. 17–54, 2004.
[26] K. Zhu, K. L. Kraemer, and S. Xu, "The Process of Innovation Assimilation by Firms in Different Countries: A Technology Diffusion Perspective on E-Business," Manage. Sci., vol. 52, no. 10, pp. 1557–1576, Oct. 2006.
[27] H.-M. Lai, I.-C. Lin, and L.-T. Tseng, "High-level managers' considerations for RFID adoption in hospitals: an empirical study in Taiwan.," J. Med. Syst., vol. 38, no. 2, p. 3, Feb. 2014.
[28] M. Swink, "Technological Innovativeness as a Moderator of New Product Design Integration and Top Management Support," J. Prod. Innov. Manag., vol. 17, no. 3, pp. 208–220, May 2000.
[29] C.-H. Yeh, G.-G. Lee, and J.-C. Pai, "Using a technology-organization-environment framework to investigate the factors influencing e-business information technology capabilities," Inf. Dev., Feb. 2014.
[30] A. Jeyaraj, J. W. Rottman, and M. C. Lacity, "A review of the predictors, linkages, and biases in IT innovation adoption research," J. Inf. Technol., vol. 21, no. 1, pp. 1–23, Jan. 2006.
[31] B. Ramdani, D. Chevers, and D. a. Williams, "SMEs' adoption of enterprise applications: A technology-organisation-environment model," J. Small Bus. Enterp. Dev., vol. 20, no. 4, pp. 735–753, 2013.
[32] A. N. Riyadh, S. Akter, and N. Islam, "The Adoption of E-banking in Developing Countries : A Theoretical Model for SMEs," Int. Rev. Bus. Res. Pap., vol. 5, no. 6, pp. 212–230, 2009.
[33] M. N. O. Sadiku, S. M. Musa, and O. D. Momoh, "Cloud Computing: Opportunities and Challenges," IEEE Potentials, vol. 33, no. 1, pp. 34–36, Feb. 2014.
[34] P. Gupta, a. Seetharaman, and J. R. Raj, "The usage and adoption of cloud computing by small and medium businesses," Int. J. Inf. Manage., vol. 33, no. 5, pp. 861–874, Oct. 2013.
[35] I. Son, D. Lee, J. Lee, and Y. Chang, "Understanding the impact of it service innovation on firm performance: The case of Cloud Computing," in PACIS 2011 Proceedings, 2011.
[36] S. Srinivasan, "Basic Cloud Computing Types," in Cloud Computing Basics, New York, NY: Springer New York, 2014, pp. 17–41.
[37] A. Khajeh-Hosseini, I. Sommerville, and I. Sriram, "Research challenges for enterprise cloud computing," arXiv Prepr. arXiv1001.3257, 2010.
[38] S. C. Misra and A. Mondal, "Identification of a company's suitability for the adoption of cloud computing and modelling its corresponding Return on Investment," Math. Comput. Model., vol. 53, no. 3–4, pp. 504–521, Feb. 2011.
[39] P. Ifinedo, "Internet/e-business technologies acceptance in Canada's SMEs: an exploratory investigation," Internet Res., vol. 21, no. 3, pp. 255–281, Jan. 2011.
[40] E. M. Rogers, Diffusion of innovations, 5th ed. New York, 2003.
[41] J. Fowler and J. Floyd, Survey research methods. Sage publications, 2008.
[42] I. Brace, Questionnaire design : how to plan, structure and write survey material for effective market research, 3rd ed. London N1 9JN: Kogan Page Limited, 2013.
[43] J. H. McDonald, HANDBOOK OF BIOLOGICAL STATISTICS. Baltimore: Sparky House, 2009.

[44] H. P. Borgman, B. Bahli, H. Heier, and F. Schewski, “Cloudrise: Exploring Cloud Computing Adoption and Governance with the TOE Framework,” in 2013 46th Hawaii International Conference on System Sciences, 2013, pp. 4425–4435.

[45] Y. Alshamaila, S. Papagiannidis, and F. Li, “Cloud computing adoption by SMEs in the north east of England: A multi-perspective framework,” J. Enterp. Inf. Manag., vol. 26, no. 3, pp. 250–275, 2013.

[46] C. Low, Y. Chen, and M. Wu, “Understanding the determinants of cloud computing adoption,” Ind. Manag. Data Syst., vol. 111, no. 7, pp. 1006–1023, 2011.

[47] R. Alghamdi, S. Drew, and S. Alkhalaf, “Government Initiatives : The Missing Key for E-commerce Growth in KSA,” World Acad. Sci. Eng. Technol., vol. 77, pp. 772–775, 2011.

[48] R. AlGhamdi, A. Nguyen, J. Nguyen, and S. Drew, “Factors influencing E-commerce Adoption by Retailers in Saudi Arabia. arXiv preprint,” Int. J. Electron. Commer. Stud., vol. 3, no. 1, pp. 83–100, 2012.

INTENTIONAL BLANK

IMAGE TYPE WATER METER CHARACTER RECOGNITION BASED ON EMBEDDED DSP

LIU Ying[1], HAN Yan-bin[2] and ZHANG Yu-lin[3]

School of Information Science and Engineering,
University of Jinan, Jinan 250022, PR China
75587050@qq.com[1]
87685630@qq.com[2]
ise_zhangyl@ujn.edu.cn[3]

ABSTRACT

In the paper, we combined DSP processor with image processing algorithm and studied the method of water meter character recognition. We collected water meter image through camera at a fixed angle, and the projection method is used to recognize those digital images. The experiment results show that the method can recognize the meter characters accurately and artificial meter reading is replaced by automatic digital recognition, which improves working efficiency.

KEYWORDS

DSP, water meter recognition, image processing & projection method

1. INTRODUCTION

In recent years, with the progress of society and science technology, it is quite common that manual work had been replaced by machines. In our daily life, water consumption statistics were mostly gotten in the form of artificial meter reading. This method not only increased the workload and difficulty of work, but also had some obvious drawbacks. The use of automatic meter reading systems not only overcome the insufficiency of manual meter reading, but also reduced the practical resources and save time.

Based on DSP chip, this study collected water meter image through the camera on hardware system. Firstly, we achieved the purpose of automatic identification through a series of image processing technology. Lastly, the designed algorithm was transplanted into hardware devices and tested. (As shown in figure 1) The advantages of the study are simply structure with high recognition rate and low cost. The identification process is mainly divided into two parts: image processing and water meter characters recognition.

David C. Wyld et al. (Eds) : CSEN, AISO, NCWC, SIPR - 2015
pp. 69–76, 2015. © CS & IT-CSCP 2015 DOI : 10.5121/csit.2015.51407

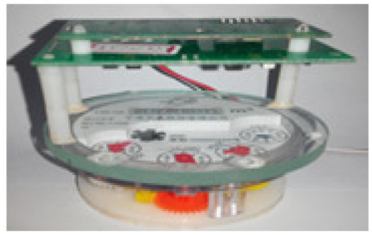


Figure 1 hardware test

2. RELATED WORK

Image processing is an important task, and it directly affects the recognition of the digital image. In the techniques of image processing, the main techniques that related to this study include image filtering, image enhancement, edge detection, image segmentation and image binarization, etc.

For example, based on the quality of image, image segmentation is divided into many concrete methods. In literature 1, it segmented digital image of water meter through counting the number of pixels in each row and column. In literature 2, it segmented digital image of water meter through vertical projection and horizontal projection. As for the image binarization, it is divided into fixed threshold method and adaptive threshold method. The literature [2-4] adopted adaptive threshold method. The literature 2 used cv adaptive threshold that was included in Opencv. The literature 3 used OTSU algorithm. And the literature 4 adopted three adaptive threshold methods. In the method of digital recognition, the literature [5, 6] adopted neural network recognition method [7, 8]. The literature 9 adopted template matching method to recognition. And the literature 10 improved template matching and used it to recognize.

In this paper, the methods that we used in image processing included image segmentation, image binarization, and image filtering. In the process of digital meter identification, we adopted the method of projection. Specific implementation process is shown in figure 2.

3. WATER METER IMAGE PRE-TREATMENT

Water meter image pre-processing is important to the character recognition. Image pre-treatment can significantly improve the quality of image and reduce noise. It is helpful for the extraction of image characteristics

3.1. Image segmentation

Because the images that were used in this study were collected through the installed cameras, the size of the image was relatively fixed. The collected water meter images had obvious border (shown in figure 3), so we segmented those pictures at a fixed point, and extracted the effective area (shown in figure 4).

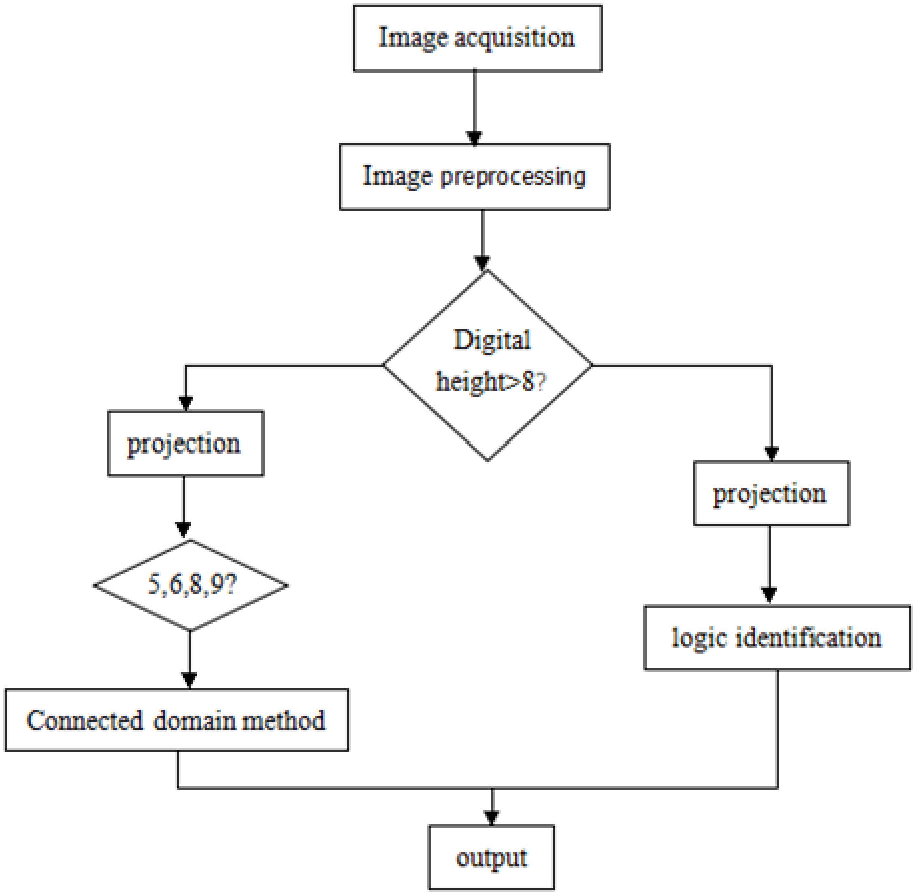


Figure 2 digital meter processing

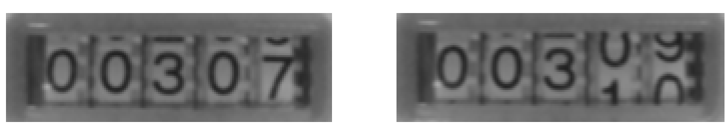


Figure 3 camera collections of pictures

Figure 4 image segmentation

3.2. Image binarization

The role of image binarization is to make the image black and white obviously. Binarization method is divided into fixed threshold method and adaptive threshold method. It is because the adaptability of fixed threshold method is poor that we generally use the adaptive threshold method.

In the study, the OTSU algorithm had simple calculation and it was not affected by the image brightness and contrast, so we used OTSU algorithm. After segmenting, the image was divided into five areas. We used the OTSU algorithm in the five areas separately for local binarization. The result is shown in figure 5.

Figure 5 the effect after binarization

3.3. Image filtering

Filtering can eliminate the noise points in the image. For binary image, we eliminate the image boundary, and then remove the small area into the isolated pixels. The result is shown in figure 6.

Figure 6 after filtering

4. DIGITAL SYMBOL RECOGNITION

There are many methods of digital symbol recognition, such as template matching, neural network recognition, statistical method and fuzzy theory method, etc.

In the research, digital symbol recognition is divided into full-word recognition, half-word recognition and logic recognition. Projection method was used in this paper. This method makes the digital project, and then, matches the projection curve with template curve. Finally, it outputs the number which has the minimum error. Because numbers 5, 6, 8 and 9 have higher similarity in this method, connected domain method was used in the two - times recognition. The connected domain method uses the number of connected domain and the location of connected domain to recognition.

Figure 7 is the projection curve of 0-9. As for every number, I selected four samples. From figure 7, we can see that the numbers 5, 6, 8 and 9 have similar characteristics. Other digital have obvious characteristics and the projection curve at the same digital has good consistency. Therefore, projection method has good ability to recognize. As for the problem that the projection method is not applicable in the numbers 5, 6, 8 and 9, we put forward the connected domain method. The method can solve the problem.

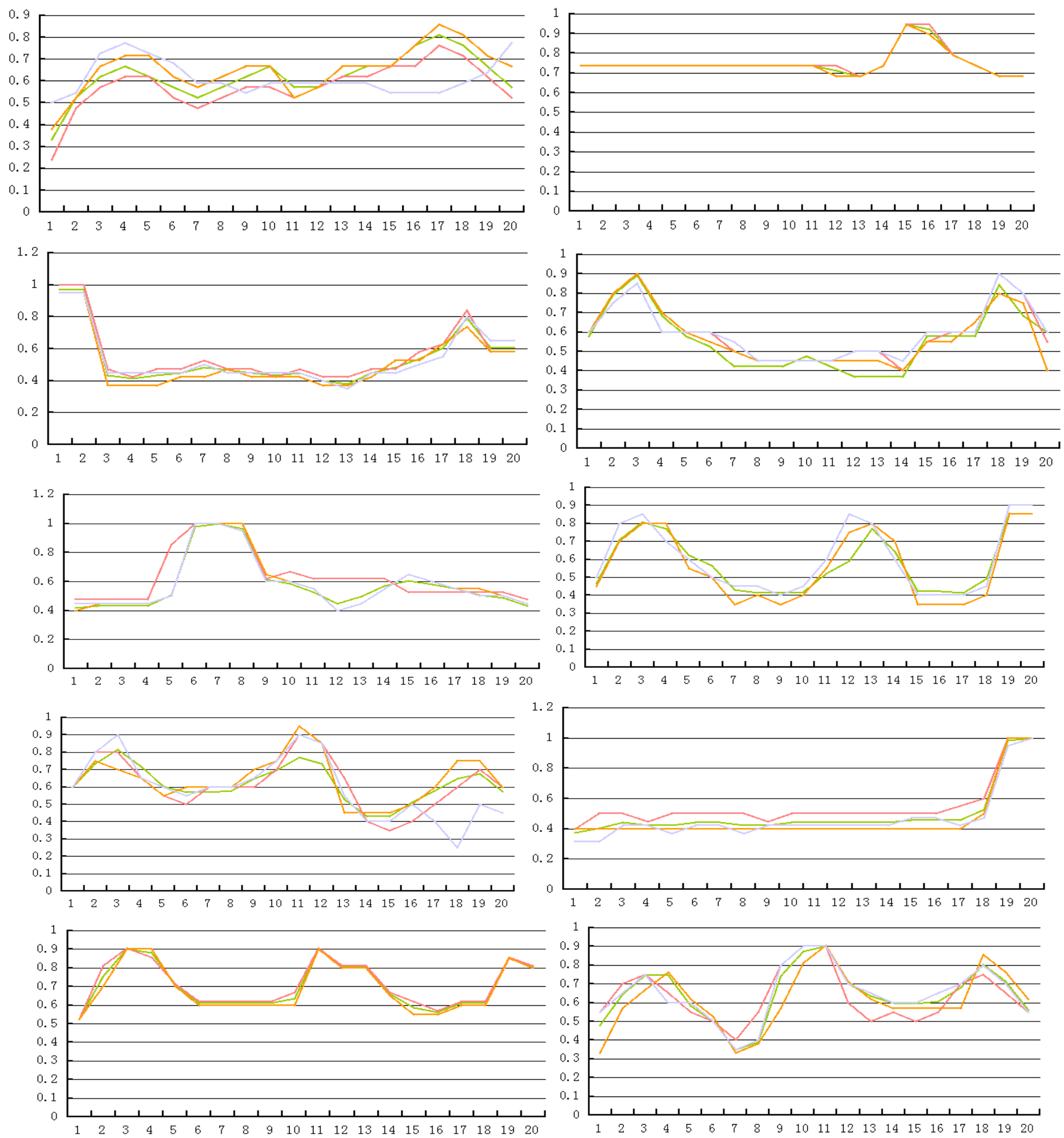


Figure 7 projection graph of 0-9

5. HALF-WORD RECOGNITION

In the above recognition method, the accuracy of full-word recognition is close to 100%. As for the half-word recognition, it was always the difficulty of the study. The common methods are template matching method, and neural network recognition method, etc. Literature 11 was studied specifically for half-word recognition. The half-word is composed of one long and one short, the algorithm is just calculated the longer half-word, and the shorter one was not considered.

In my paper, I improved the template matching method based on the projection method of full-word recognition. (The image of half-word is shown in figure 8) I reserved the long part and the short pat, and then, I used the method of projection on them respectively. Finally, I analysed the matching result and recognized the digital according to the logical continuity.

Figure 8 half-word image

Table 1 is the matched data of the half-word in figure 8. In matched data in the above part, the number 3 has the minimum error. In matched data of the below part, the number 4 has the minimum error. Because of the number 3 and the number 4 have continuity in logic, they can recognize correctly.

Table 1 the matched data of the half-word in figure 8

match object	0	1	2	3	4	5	6	7	8	9
error (above)	3.5	5.8	4.4	1.6	4.9	4.0	3.0	5.4	3.6	4.1
error (below)	7.2	6.4	6.2	5.3	1.5	6.6	6.3	5.7	5.7	6.5

6. LOGIC IDENTIFY

Aiming at several half-word in figure 9, there is needed to adopt logical identification. When the image appears more than one half-word, if we only recognize the longer half-word, there will be a cross recognition (as shown in figure 9, those images may be recognized to 310 and 300) and cause errors. So we need to combine the continuity of digital to recognize. The method not only saves the calculation time, but also improves the recognition accuracy.

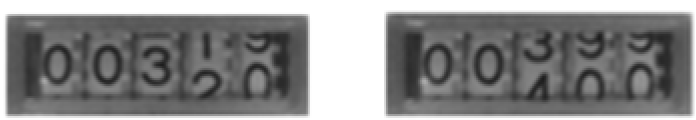


Figure 9 several half-words

The recognition results of some typical half-word are shown in figure 10. The above of the image is original image, the lower left corner is the processed image, and the lower right is recognition results.

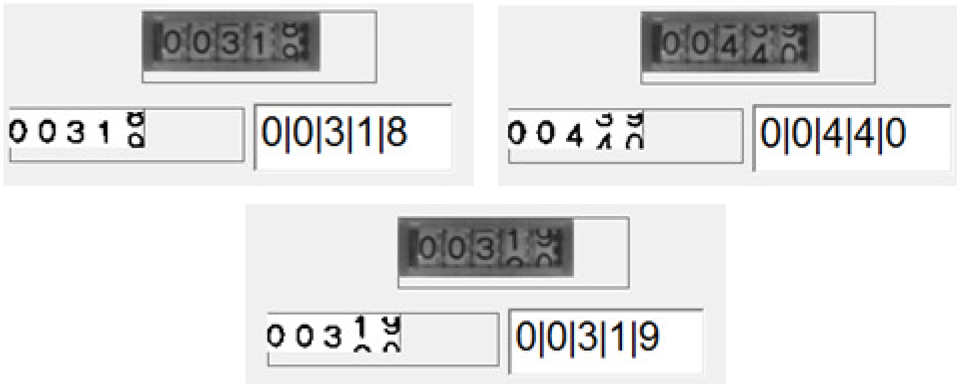


Figure 10 recognition results

7. CONCLUSIONS

This study used the projection method as the main means of digital recognition. In addition, we put forward the connected domain method and logical identification method in half-word. In this paper, the scheme of automatic recognition of digital water meter has good results, especially for the difficulty recognition of half-word. What is more, this study adopts the DSP chip. Through combination with embedded technology, making this study has practical application value. Through testing, the method that mentioned in this paper obtained a good accuracy. The recognition accuracy can reach above 95%. The method is simple and effective, and can be used to replace artificial meter reading. It overcomes the disadvantages of manual meter reading and has good practical value.

ACKNOWLEDGEMENTS

This article is completed under the guidance of professor ZHANG and professor HAN. In addition, my classmates also gave me a lot of help. What is more, University of Jinan provided a good learning environment for us.

REFERENCES

[1] LI Yun-dong & SUN Chang-qing, (2013) “Research on Real-Time Water Meter Recognition System Based on BF Neural Network”, Computer Measurement & Control, Vol. 21, No. 5, pp1330-1332.

[2] GAO Ju & YE Hua,(2013)”An effective two-times recognition algorithm of meter digital image”, JOURNAL OF SOUTHEAST UNIVERSITY(Natural Science Edition), Vol. 7, No.1, pp153-157.

[3] XI Run-ping, ZHAO Rong-chun, YUE Si-cong & YI Fei, (2007) “An Automatic Reading and Recognition System for Water Meter Based on DSP and Image Processing”, Measurement & Control Technology, Vol. 26, No. 5, pp21-25.

[4] YANG Xin-yi, LV Wei-tao, YANG Jun, ZHANG Ke, MA Ying, YAO Wen & LI Qing-yong (2014), “Three kinds of threshold methods in the application of lightning channel image recognition”, JOURNAL OF APPLIED METEORLOLGICAL SCIENCE, Vol. 25, No. 4, pp427-435.

[5] FAN Chong-qing, (2004) “Using the BP neural network to identify digital image”, Journal of Higher Correspondence Education (Natural Sciences), Vol. 17, No. 6, pp46-49.

[6] ZHANG Yi-cheng, CHEN Xin, YANG Hong-jun, ZHANG Yu-jian (2013), “Neural Network Digital Characters Recognition Based on the Combined Feature”, Computer System & Applications, Vol. 22, No. 3, pp113-116.

[7] ZHOU Yu, ZHU An-fu, ZHOU Lin & QIAN Xu,(2012) “Sample data selection method for neural network classifiers”, J.Huazhong Univ. of Sci. & Tech. (Natural Science Edition), Vol. 40, No. 6, pp39-43.

[8] WANG Ting, JIANG Wen-hui & XIAO Nan-feng (2011), “Numerical recognition based on improved BP neural network”, Electronic Design Engineering, Vol.19, No.3, pp108-112.

[9] LI Zhi-rong, YANG Dan & ZHOU Qi (2009), “Digital recognition based on template matching”, NATURAL SCIENCES JOURNAL OF HARBIN NORMAL UNIVERSITY, Vol.25, No.4, pp86-95.

[10] ZOU Ming-ming & LU Di,(2010) “Recognition algorithm of car license plate characters based on modified template match”, Foreign Electronic Measurement Technology, Vol. 29, No.1, pp59-80.

[11] JIN Jing-xiao & XIN Kun-lun,(2015) “Application of improved half-word recognition method in automatic meter reading system”, WATER TECHNOLOGY, Vol.9 No.1, pp43-45.

AUTHORS

LIU Ying (1991-), female, master, research in switched reluctance motor, pattern recognition, image processing.

HAN Yan-bin (1979-), male, associate professor, research in pattern recognition, image processing.

ZHANG Yu-lin (1972-), corresponding author, male, professor, the research direction for embedded system, computer architecture.

HIERARCHICAL DEEP LEARNING ARCHITECTURE FOR 10K OBJECTS CLASSIFICATION

Atul Laxman Katole, Krishna Prasad Yellapragada, Amish Kumar Bedi, Sehaj Singh Kalra and Mynepalli Siva Chaitanya

Samsung R&D Institute India - Bangalore, Bagmane Constellation Business Park, Doddanekundi Circle, Bangalore, India

ABSTRACT

Evolution of visual object recognition architectures based on Convolutional Neural Networks & Convolutional Deep Belief Networks paradigms has revolutionized artificial Vision Science. These architectures extract & learn the real world hierarchical visual features utilizing supervised & unsupervised learning approaches respectively. Both the approaches yet cannot scale up realistically to provide recognition for a very large number of objects as high as 10K. We propose a two level hierarchical deep learning architecture inspired by divide & conquer principle that decomposes the large scale recognition architecture into root & leaf level model architectures. Each of the root & leaf level models is trained exclusively to provide superior results than possible by any 1-level deep learning architecture prevalent today. The proposed architecture classifies objects in two steps. In the first step the root level model classifies the object in a high level category. In the second step, the leaf level recognition model for the recognized high level category is selected among all the leaf models. This leaf level model is presented with the same input object image which classifies it in a specific category. Also we propose a blend of leaf level models trained with either supervised or unsupervised learning approaches. Unsupervised learning is suitable whenever labelled data is scarce for the specific leaf level models. Currently the training of leaf level models is in progress; where we have trained 25 out of the total 47 leaf level models as of now. We have trained the leaf models with the best case top-5 error rate of 3.2% on the validation data set for the particular leaf models. Also we demonstrate that the validation error of the leaf level models saturates towards the above mentioned accuracy as the number of epochs are increased to more than sixty. The top-5 error rate for the entire two-level architecture needs to be computed in conjunction with the error rates of root & all the leaf models. The realization of this two level visual recognition architecture will greatly enhance the accuracy of the large scale object recognition scenarios demanded by the use cases as diverse as drone vision, augmented reality, retail, image search & retrieval, robotic navigation, targeted advertisements etc.

KEYWORDS

Convolutional Neural Network [CNN], Convolutional Deep Belief Network [CDBN], Supervised & Unsupervised training

David C. Wyld et al. (Eds) : CSEN, AISO, NCWC, SIPR - 2015
pp. 77–93, 2015. © CS & IT-CSCP 2015 DOI : 10.5121/csit.2015.51408

1. INTRODUCTION

Deep learning based vision architectures learn to extract & represent visual features with model architectures that are composed of layers of non-linear transformations stacked on top of each other [1]. They learn high level abstractions from low level features extracted from images utilizing supervised or unsupervised learning algorithms. Recent advances in training CNNs with gradient descent based backpropagation algorithm have shown very accurate results due to inclusion of rectified linear units as nonlinear transformation [2]. Also extension of unsupervised learning algorithms that train deep belief networks towards training convolutional networks have exhibited promise to scale it to realistic image sizes [4]. Both the supervised and unsupervised learning approaches have matured and have provided architectures that can successfully classify objects in 1000 & 100 categories respectively. Yet both the approaches cannot be scaled realistically to classify objects from 10K categories.

The need for large scale object recognition is ever relevant today with the explosion of the number of individual objects that are supposed to be comprehended by artificial vision based solutions. This requirement is more pronounced in use case scenarios as drone vision, augmented reality, retail, image search & retrieval, industrial robotic navigation, targeted advertisements etc. The large scale object recognition will enable the recognition engines to cater to wider spectrum of object categories. Also the mission critical use cases demand higher level of accuracy simultaneously with the large scale of objects to be recognized.

In this paper, we propose a two level hierarchical deep learning architecture that achieves compelling results to classify objects in 10K categories. To the best of our knowledge the proposed method is the first attempt to classify 10K objects utilizing a two level hierarchical deep learning architecture. Also a blend of supervised & unsupervised learning based leaf level models is proposed to overcome labelled data scarcity problem. The proposed architecture provides us with the dual benefit in the form of providing the solution for large scale object recognition and at the same time achieving this challenge with greater accuracy than being possible with a 1-level deep learning architecture.

2. RELATED WORKS

We have not come across any work that uses 2-level hierarchical deep learning architecture to classify 10K objects in images. But object recognition on this large scale using shallow architectures utilizing SVMs is discussed in [5]. This effort presents a study of large scale categorization with more than 10K image classes using multi-scale spatial pyramids (SPM) [14] on bag of visual words (BOW) [13] for feature extraction & Support Vector Machines (SVM) for classification.

This work creates ten different datasets derived from ImageNet each with 200 to 10,000 categories. Based on these datasets it outlines the influence on classification accuracy due to different factors like number of labels in a dataset, density of the dataset and the hierarchy of labels in a dataset. The methods are proposed which provide extended information to the classifier on the relationship between different labels by defining a hierarchical cost. This cost is calculated as the height of the lowest common ancestor in WordNet. Classifiers trained on loss function using the hierarchical cost can learn to differentiate and predict between similar categories when compared to those trained on 0-1 loss. The error rate for the classification of entire 10K categories is not conclusively stated in this work.

3. Problem Statement

Supervised learning based deep visual recognition CNN architectures are composed of multiple convolutional stages stacked on top of each other to learn hierarchical visual features [1] as captured in Figure 1. Regularization approaches such as stochastic pooling, dropout, data augmentation have been utilized to enhance the recognition accuracy. Recently the faster convergence of these architectures is attributed to the inclusion of Rectified Linear Units [ReLU] nonlinearity into each of the layer with weights. The state of the art top 5 error rate reported is 4.9% for classification into 1K categories [6] that utilizes the above mentioned generic architectural elements in 22 layers with weights.

Unsupervised learning based architecture model as convolutional DBN learns the visual feature hierarchy by greedily training layer after layer. These architectures have reported accuracy of 65.4% for classifying 101 objects [4].

Both the architectures are not yet scaled for classification of 10K objects. We conjecture that scaling a single architecture is not realistic as the computations will get intractable if we utilize deeper architectures.

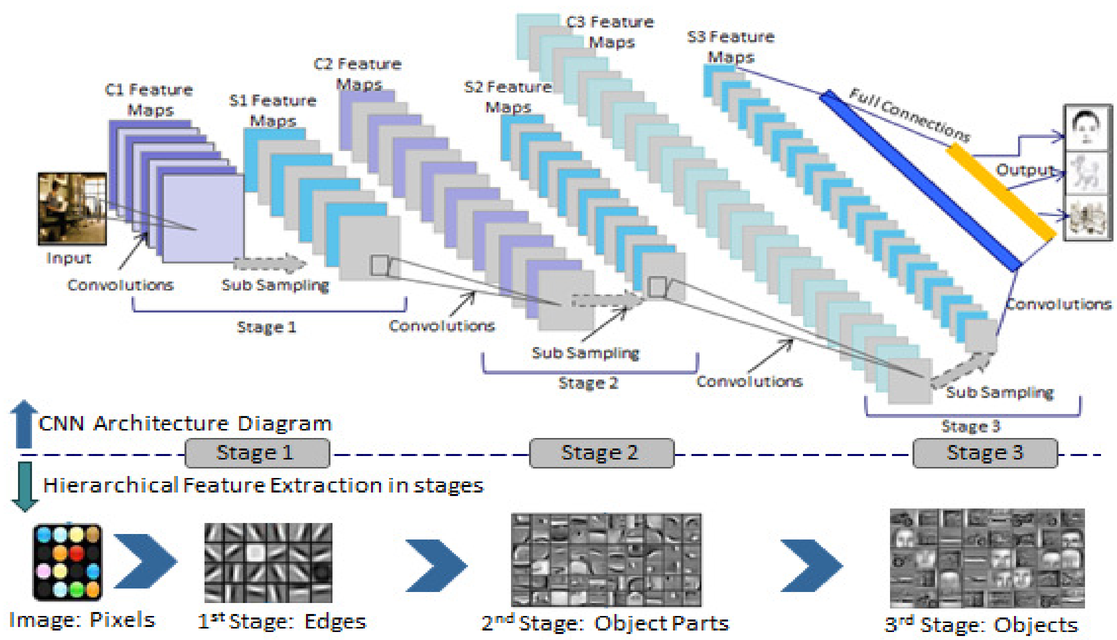


Figure 1. Learning hierarchy of visual features in CNN architecture

4. PROPOSED METHOD

We employ divide & conquer principle to decompose the 10K classification into root & leaf level distinct challenges. The proposed architecture classifies objects in two steps as captured below:

1. Root Level Model Architecture: In the first step the root i.e. the first level in architectural hierarchy recognizes high level categories. This very deep vision architectural model with 14 weight layers [3] is trained using stochastic gradient descent [2]. The architectural elements are captured in the table 1.

2. Leaf Level Model Architecture: In the second step, the leaf level recognition model for the recognized high level category is selected among all the leaf models. This leaf level model is presented with the same input object image which classifies it in a specific category. The leaf level architecture in the architectural hierarchy recognizes specific objects or finer categories. This model is trained using stochastic gradient descent [2]. The architectural elements are captured in the table 2.

CDBN based leaf level models can be trained with unsupervised learning approach in case of scarce labelled images [4]. This will deliver a blend of leaf models trained with supervised & unsupervised approaches. In all a root level model & 47 leaf level models need to be trained. We use ImageNet10K dataset [5], which is compiled from 10184 synsets of the Fall-2009 release of ImageNet. Each leaf node has at least 200 labelled images which amount to 9M images in total.

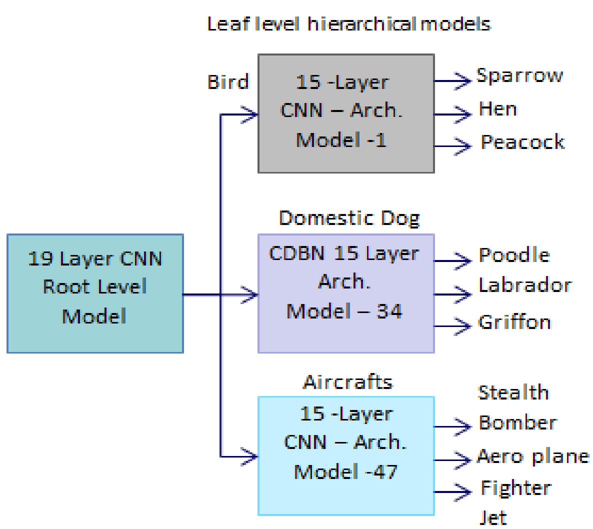


Figure 2. Two Levels – Hierarchical Deep Learning Archtecture

5. SUPERVISED TRAINING

In vision the low level features [e.g. pixels, edge-lets, etc.] are assembled to form high level abstractions [e.g. edges, motifs] and these higher level abstractions are in turn assembled to form further higher level abstractions [e.g. object parts, objects] and so on. Substantial number of the recognition models in our two-level hierarchical architecture is trained utilizing supervised training. The algorithms utilized for this method are referred to as error-back propagation algorithm. These algorithms require significantly high number of labelled training images per object category in its data set.

5.1. CNN based Architecture

CNN is a biologically inspired architecture where multiple trainable convolutional stages are stacked on the top of each other. Each CNN layer learns feature extractors in the visual feature hierarchy and attempts to mimic the human visual system feature hierarchy manifested in different areas of human brain as V1 & V2 [10]. Eventually the fully connected layers act as a feature classifier & learn to classify the extracted features by CNN layers into different categories or objects. The fully connected layers can be likened to the V4 area of the brain which classifies the hierarchical features as generated by area V2.

The root level & the leaf level CNN models in our architecture are trained with supervised gradient descent based backpropagation method. In this learning method, the cross entropy objective function is minimized with the error correction learning rule/mechanism. This mechanism computes the gradients for the weight updates of the hidden layers by recursively computing the local error gradient in terms of the error gradients of the next connected layer of neurons. By correcting the synaptic weights for all the free parameters in the network, eventually the actual response of the network is moved closer to the desired response in statistical sense.

Table 1. CNN architecture layers [L] with architectural elements for root level visual recognition architecture

Layer No.	Type	Input Size	#kernels
1	Convolutional	225 x 225 x 3	3 x 3 x 3
2	Max Pool	223 x 223 x 64	3 x 3
3	Convolutional	111 x 111 x 64	3 x 3 x 64
4	Convolutional	111 x 111 x 128	3 x 3 x 128
5	Max Pool	111 x 111 x 128	3 x 3
6	Convolutional	55 x 55 x 128	3 x 3 x 128
7	Convolutional	55 x 55 x 256	3 x 3 x 256
8	Max Pool	55 x 55 x 256	3 x 3
9	Convolutional	27 x 27 x 256	3 x 3 x 256
10	Convolutional	27 x 27 x 384	3 x 3 x 384
11	Convolutional	27 x 27 x 384	3 x 3 x 384
12	Max Pool	27 x 27 x 384	3 x 3
13	Convolutional	13 x 13 x 384	3 x 3 x 384
14	Convolutional	13 x 13 x 512	3 x 3 x 512
15	Convolutional	13 x 13 x 512	3 x 3 x 512
16	Max Pool	13 x 13 x 512	3 x 3
17	Convolutional	7 x 7 x 512	1 x 1 x 512
18	Full-Connect	12544	4096
19	Full-Connect	4096	2048
20	Full-Connect	2048	128
	128-Softmax	128	

5.2. Architectural Elements

Architectural elements for the proposed architecture are:

- *Enhanced Discriminative Function*: We have chosen deeper architectures & smaller kernels for the root & leaf models as they make the objective function more discriminative. This can be interpreted as making the training procedure more difficult by making it to choose the feature extractors from higher dimensional feature space.

- *ReLU Non-linearity:* We have utilized ReLU nonlinearities as against sigmoidal i.e. non-saturating nonlinearities in each layer as it reduces the training time by converging upon the weights faster [2].

- *Pooling*: The output of convolutional-ReLU combination is fed to a pooling layer after alternative convolutional layers. The output of the pooling layer is invariant to the small changes in location of the features in the object. The pooling method used is either max-pooling OR stochastic pooling. Max Pooling method averages the output over the neighborhood of the neurons where-in the poling neighborhoods can be overlapping or non-overlapping. In majority of the leaf models we have used Max Pooling approach with overlapping neighborhoods.

  Alternatively we have also used Stochastic Pooling method when training for few models. In Stochastic Pooling the output activations of the pooling region are randomly picked from the activations within each pooling region, following multinomial distribution. This distribution is computed from the neuron activation within the given region [12]. This approach is hyper parameter free. The CNN architecture for stochastic pooling technique is captured in table 3.

- Dropout: With this method the output of each neuron in fully connected layer is set to zero with probability 0.5. This ensures that the network samples a different architecture when a new training example is presented to it. Besides this method enforces the neurons to learn more robust features as it cannot rely on the existence of the neighboring neurons.

Table 2. CNN architecture layers [L] with architectural elements for leaf level visual recognition architecture with Max pooling strategy

Layer No.	**Type**	**Input Size**	**#kernels**
1	Convolutional	225 x 225 x 3	7 x 7 x 3
2	Max Pool	111 x 111 x 64	3 x 3
3	Convolutional	55 x 55 x 64	3 x 3 x 64
4	Convolutional	55 x 55 x 128	3 x 3 x 128
5	Max Pool	55 x 55 x 128	3 x 3
6	Convolutional	27 x 27 x 128	3 x 3 x 128
7	Convolutional	27 x 27 x 256	3 x 3 x 256
8	Max Pool	27 x 27 x 256	3 x 3
9	Convolutional	13 x 13 x 256	3 x 3 x 256
10	Convolutional	13 x 13 x 384	3 x 3 x 384
12	Max Pool	13 x 13 x 384	3 x 3
13	Convolutional	7 x 7 x 384	1 x 1 x 384
14	Full-Connect	6272	2048
15	Full-Connect	2048	2048
16	Full-Connect	2048	256
	256-Softmax	256	

Table 3. CNN architecture layers [L] with architectural elements for leaf level visual recognition architecture with stochastic pooling strategy

Layer No.	Type	Input Size	#kernels
1	Convolutional	225 x 225 x 3	7 x 7 x 3
2	Stochastic Pool	111 x 111 x 64	3 x 3
3	Convolutional	55 x 55 x 64	3 x 3 x 64
4	Convolutional	55 x 55 x 128	3 x 3 x 128
5	Stochastic Pool	55 x 55 x 128	3 x 3
6	Convolutional	27 x 27 x 128	3 x 3 x 128
7	Convolutional	27 x 27 x 256	3 x 3 x 256
8	Stochastic Pool	27 x 27 x 256	3 x 3
9	Convolutional	13 x 13 x 256	3 x 3 x 256
10	Convolutional	13 x 13 x 384	3 x 3 x 384
12	Stochastic Pool	13 x 13 x 384	3 x 3
13	Convolutional	7 x 7 x 384	1 x 1 x 384
14	Full-Connect	6272	2048
15	Full-Connect	2048	2048
16	Full-Connect	2048	256
	256-Softmax	256	

5.3. Training Process

We modified libCCV open source CNN implementation to realize the proposed architecture which is trained on NVIDIA GTX™ TITAN GPUs. The root & leaf level models are trained using stochastic gradient descent [2].

The leaf models are trained as batches of 10 models per GPU on the two GPU systems simultaneously. The first 4 leaf models were initialized and trained from scratch for 15 epochs with learning rate of 0.01, and momentum of 0.9. The rest of the leaf models are initialized from the trained leaf models and trained with learning rate as 0.001. The root model has been trained for 32 epochs with learning rate of 0.001, after having been initialized from a similar model trained on ImageNet 1K dataset.

It takes 10 days for a batch of 20 leaf models to train for 15 epochs. Currently the root model and 25/47 leaf models have been trained in 5 weeks. Full realization of this architecture is in progress and is estimated to conclude by second week of September'15

6. UNSUPERVISED TRAINING

Statistical Mechanics has inspired the concept of unsupervised training fundamentally. Specifically statistical mechanics forms the study of macroscopic equilibrium properties of large system of elements starting from the motion of atoms and electrons. The enormous degree of freedom as necessitated by statistical mechanics foundation makes the use of probabilistic methods to be the most suitable candidate for modelling features that compose the training data sets [9].

6.1. CDBN based Architecture

The networks trained with statistical mechanics fundamentals model the underlying training dataset utilizing Boltzmann distribution. To obviate the painfully slow training time as required to train the Boltzmann machines, multiple variants of the same have been proposed where the Restricted Boltzmann Machine [RBM] is the one that has provided the best possible modelling capabilities in minimal time. The resulting stacks of RBM layers are greedily trained layer by layer [4] resulting in the Deep Belief Networks [DBN] that successfully provides the solution to image [1- 4], speech recognition [8] and document retrieval problem domains.

DBN can be described as multilayer generative models that learn hierarchy of non-linear feature detectors. The lower layer learns lower level features which feeds into the higher level and help them learn complex features. The resulting network maximizes the probability that the training data is generated by the network. But DBN has its own limitations when scaling to realistic image sizes [4]. First difficulty is to be computationally tractable with increasing image sizes. The second difficulty is faced with lack of translational invariance when modelling images.

To scale DBN for modelling realistic size images the powerful concept of Convolutional DBN [CDBN] had been introduced. CDBN learns feature detectors that are translation invariant i.e. the feature detectors can detect the features that can be located at any location in an image.

We perform the block Gibbs sampling using conditional distribution as suggested in [4] to learn the convolutional weights connecting the visible and hidden layers where v and h are activations of neurons in visible & hidden layers respectively. Also b_j are hidden unit biases and c_i are visible unit biases. W forms the weight matrix connecting the visible and hidden layer. The Gibbs sampling is conducted utilizing (1) & (2).

$$P(h_{ij} = 1 \mid \boldsymbol{v}) = \; sigmoid((W * v)_{ij} \; + \; b_j) \qquad (1)$$

$$P(v_{ij} = 1 \mid \boldsymbol{h}) = \; sigmoid((\sum W * v)_{ij} \; + \; c_i) \qquad (2)$$

The weights such learnt give us the layers for Convolutional RBMs [CRBM]. The CRBMs can be stacked on top of each other to form CDBN. We had probabilistic Max Pooling layers after convolutional layers [4]. MLP is introduced at the top to complete the architecture. This concept is captured in Figure 3.

We train the first two layers in the leaf architecture with unsupervised learning. Later we abandon unsupervised learning and use the learnt weights in the first two layers to initialize the weights in CNN architecture. The CNN architecture weights are then fine-tuned using backpropagation method. The architecture used for training with unsupervised learning mechanism is same as captured in table 2. Also a two-level hierarchical deep learning architecture can be constructed entirely with CDBN as depicted in Figure 4.

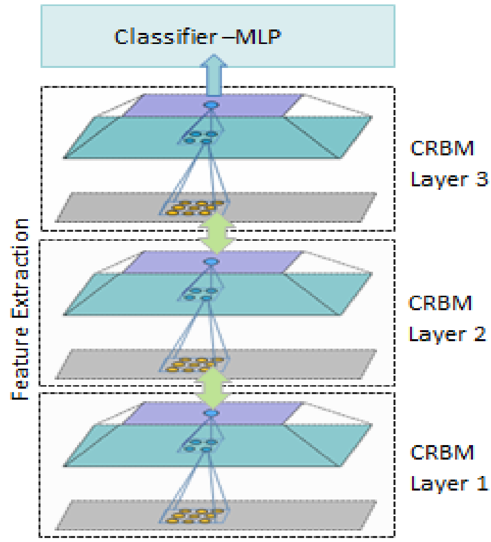


Figure 3. Convolutional DBN constructed by stacking CRBMs which are learnt one by one.

6.2. Training Process

We have used Contrastive Divergence CD-1 mechanism to train the first two layers of the architecture as specified for unsupervised learning. The updates to the hidden units in the positive phase of CD-1 step were done with sampling rather than using the real valued probabilities. Also we had used mini-batch size of 10 when training.

We had monitored the accuracy of the training utilizing –

1. Reconstruction Error: It refers to the squared error between the original data and the reconstructed data. While it does not guarantee accurate training, but during the course of training it should generally decrease. Also, any large amount of increase suggests the training is going wrong.

2. Printing learned Weights: The learned weighs needs to be eventually visualized as oriented, localized edge filters. Printing weights during training helps identify whether the weights are “approaching” that filter-like shape.

When the ratio of variance of reconstructed data to variance of input image exceeds 2, we decrease the learning rate by factor of 0.9 and reset the values of weights and hidden bias updates to ensure that weights don’t explode. The initial momentum was chosen to be 0.5 which is increased finally to 0.9. The initial learning rate is chosen to be 0.1.

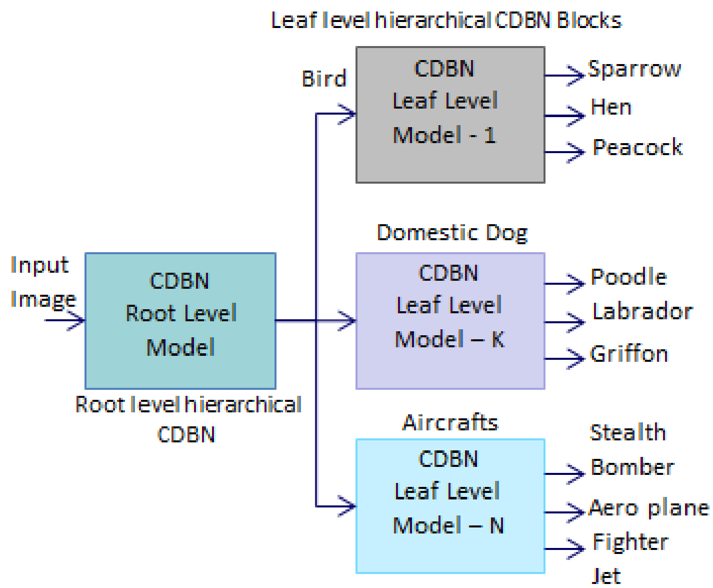


Figure 4. Proposed 2 Level Hierarchical Deep Learning Architecture constructed entirely utilizing CDBNs for classification/Recognition of 10K+ objects

7. TWO-LEVEL HIERARCHY CREATION

The 2-level hierarchy design for classification of 10K objects categories requires decision making on the following parameters –

1. Number of leaf models to be trained and

2. Number of output nodes in each leaf model.

To decide these parameters, we first build a hierarchy tree out of the 10184 synsets (classes) in ImageNet10K dataset (as described in section 7.1). Then using a set of thumb-rules (described in section 7.2), we try to split and organize all the classes into 64 leaf models, each holding a maximum of 256 classes

7.1. Building Hierarchical Tree

Using the WordNet IS A relationship, all the synsets of ImageNet10K dataset are organized into a hierarchical tree. The WordNet ISA relationship is a file that lists the parent-child relationships between synsets in ImageNet. For example a line “n02403454 n02403740” in the relationship file refers to the parent synset to be n02403454 (cow) and child synset as n02403740 (heifer). However the ISA file can relate a single child to multiple parents, i.e. heifer is also the child of another category n01321854 (young mammal). As the depth of a synset in ImageNet hierarchy has no relationship to its semantic label, we focused on building the deepest branch for a synset. We utilized a simplified method that exploits the relationship between synset ID and depth; the deeper a category *nXXX* the larger its number *XXX*. Hence we used the parent category of *heifer* as *cow*, instead of *young mammal*.

The algorithm Htree as depicted in Figure 5a & 5b is used to generate the hierarchy tree. In this paper, the results from the *ninth (9th) iteration* are used as base tree. A sample illustration is captured in Figure 6.

7.2. Thumb-rules for Building Hierarchical Tree

From the Hierarchy tree, it is evident that the dataset is skewed towards the categories like flora, animal, fungus, natural objects, instruments etc. that are at levels closer to the tree root i.e. 80% of the synsets fall under 20% of the branches.

```
Algorithm: deepest_branch
Finding the deepest branch for a synset

•  Input:
     –  synset
     –  WordNet_ISA.map
•  parents := WordNet_ISA.map[synset]
•  closest_parent := Max([XXX from nXXX of parents])
•  branch := deepest_branch(closest_parent, WordNet_ISA.map)
•  append closest_parent to branch
•  return branch
```

Figure 5a. Pseudo-code for Hierarchy Tree Generation (HTree) Algorithm

```
Algorithm: Generate Hierarchy Tree for the Synsets in ImageNet10K, using deepest_branch
algorithm.

•  Input:
     –  Synsets.list
     –  WordNet_ISA.map
•  branches := []
•  for each synset in synset.list
     –  branch := deepest_branch(synset, WordNet_ISA.map)
     –  append branch to branches
•  depths := [len(branch) for branch in branches]
•  max_depth := max(depths)
•  for iter := [max_depth .. 1]
     –  roll-up and merge leafs where depth = iter
```

Figure 5b. Pseudo-code for Hierarchy Tree Generation (HTree) Algorithm

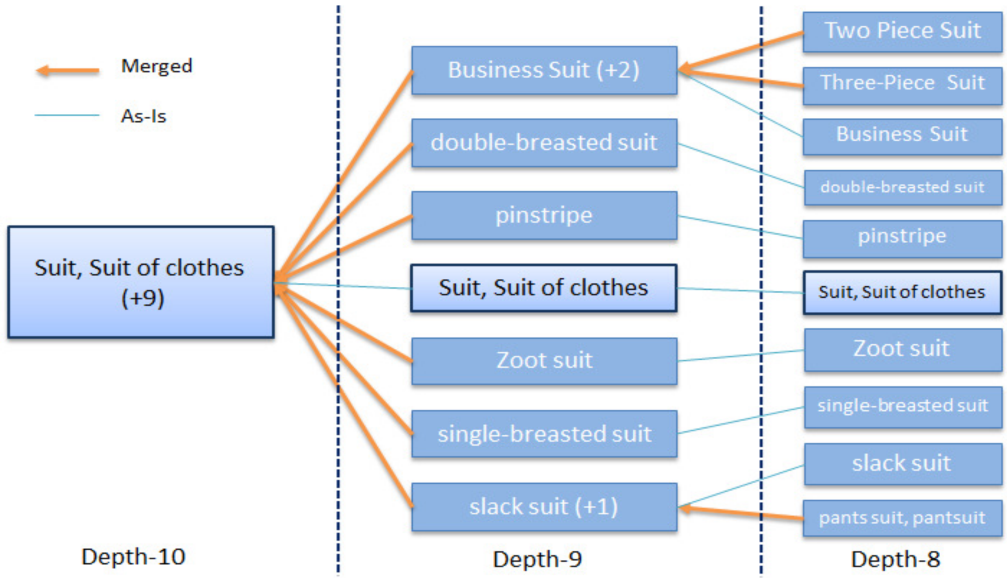


Figure 6. Hierarchy Tree Generated by HTree algorithm at iterations 8, 9 & 10 for ImageNet10K. The categories at Depth-9 are part of leaf-1 model.

Taking into account the number of models to be trained and the time & resources required for fine-tuning each model, the below thumb-rules were decided to finalize the solution parameters:

1. The ideal hierarchy will have 64 leaf models each capable of classifying 256 categories
2. The synsets for root and leaf level models have to be decided such that the hierarchy tree is as flat as possible
3. The total number of synsets in a leaf model should be less than 256 and
4. If leaf level models have more than 256 sub-categories under it, the remaining subcategories will be split or merged with another leaf.

7.3. Final Solution Parameters

The final solution parameters are as follows, 47 leaf models and with each leaf model classifying 200 ~ 256 synsets.

8. RESULTS

We formulate the root & leaf models into 2-level hierarchy. In all a root level model & 47 leaf level models need to be trained. Each leaf level model recognizes categories that range from 45 to 256 in numbers. We use ImageNet10K dataset [7], which is compiled from 10184 synsets of the Fall-2009 release of ImageNet.

Each leaf node has at least 200 labelled images which amount to 9M images in total. The top 5 error rates for 25 out of 47 leaf level models have been computed. The graph in Fig. 5 plots the top 5 errors of leaf models vis-à-vis the training epochs. We observe that when the leaf models are trained with higher number of training epochs the top 5 error decreases. The top 5 error rate for the complete 10K objects classification can be computed upon training of all the 47 models as required by the 2-level hierarchical deep learning architecture.

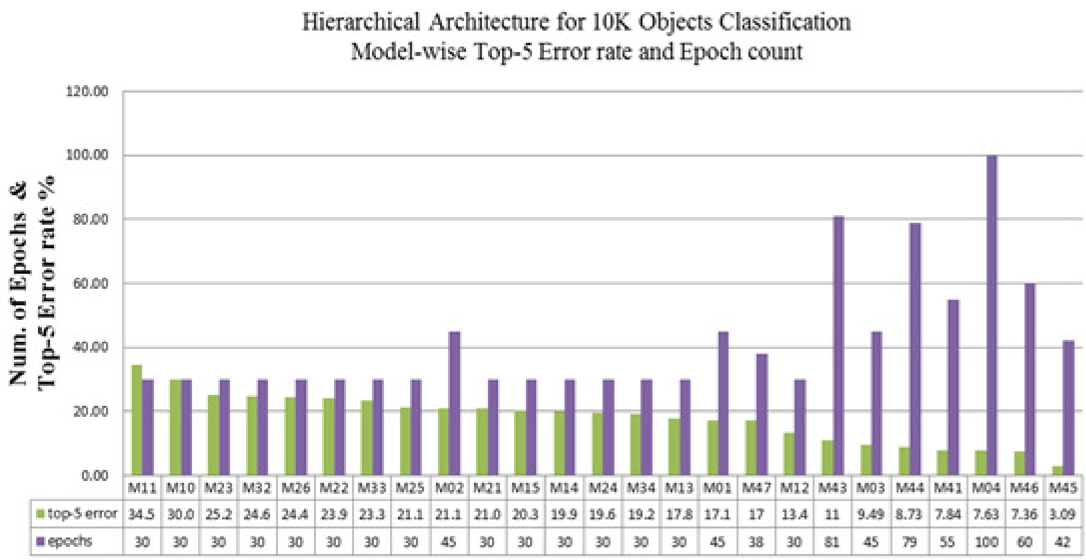


Figure 7. Graph captures the decrease in top-5 error rate with increase in increase in number of epochs when training with supervised training method

Training for classification of 10K objects with the proposed 2-level hierarchical architecture is in progress and is estimated to be completed by mid of September'15.

In this architecture, the root model & the 46/47 leaf models are based on CNN architecture and trained with supervised gradient descent.

Utilizing unsupervised learning we have trained a leaf model Leaf-4 that consists of man-made artifacts with 235 categories. The model for Leaf-4 is CDBN based architecture as described in Section 6. We have trained the first layer of this CDBN architecture with contrastive divergence (CD-1) algorithm. Later on the first layer weights are utilized to initialize the weights for leaf-4 model in supervised setting. The same are then fine-tuned with back-propagation utilizing supervised learning. The feature extractors or kernels learnt with supervised & unsupervised learning are captured in Fig. 7.

We intend to compute the top 5 error rates for categories using the algorithm as captured in the Fig. 9. Figures 10 – 13 depict the Top-5 classification results using this 2-level hierarchical deep learning architecture. From top-left, the first image is the original image used for testing. The remaining images represent the top-5 categories predicted by the hierarchical model, in descending order of their confidence.

9. CONCLUSIONS

The top-5 error rate for the entire two-level architecture is required to be computed in conjunction with the error rates of root & leaf models. The realizations of this two level visual recognition architecture will greatly simply the object recognition scenario for large scale object recognition problem. At the same time the proposed two level hierarchical deep learning architecture will help enhance the accuracy of the complex object recognition scenarios significantly that otherwise would not be possible with just 1-level architecture.

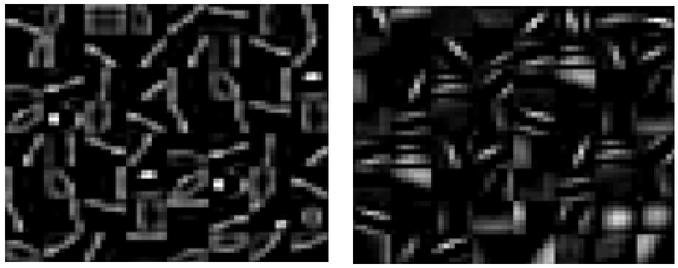


Figure 8. : The left image depicts the first layer 64 numbers of 7*7 filter weights learned by leaf-4 utilizing CD-1. The right image depicts the same first layer weights after fine-tuning with back propagation.

The trade of with the proposed 2-level architecture is the size of the hierarchical recognition model. The total size of the 2-level recognition models including the root & leaf models amounts to approximately 5 GB. This size might put constraints towards executing the entire architecture on low-end devices. The same size is not a constraint when executed with high end device or cloud based recognition where RAM size is higher. Besides we can always split the 2-level hierarchical model between device & cloud which paves the way for object recognition utilizing the novel device-cloud collaboration architectures.

The proposed 10K architecture will soon be available for classifying large scale objects. This breakthrough will help enable applications as diverse as drone vision, industrial robotic vision, targeted advertisements, augmented reality, retail, robotic navigation, video surveillance, search & information retrieval from multimedia content etc. The hierarchical recognition models can be deployed and commercialized in various devices like Smart phones, TV, Home Gateways and VR head set for various B2B and B2C use cases.

Algorithm 1: Error eval. algorithm for 10K 2-level Hierarchical architecture

---

- Input:
  - ROOT.model, LEAF[1-N].model, LID_GID.map
  - ROOT Test Data: $\{GID, RID, X\}^+$, where
    - GID is the global synset ID [1-10184]
    - RID is the root category [1-47]
    - X is the input image
- top5_error = 0
- For each test in $\{GID, RID, X\}^+$
  - rootTop5 := classify(ROOT.model, X) where rootTop5: $\{RID, RCONF\}^5$
  - For each RID in rootTop5:
    - leafTop5[RID] := classify(LEAF[RID].model, X) where leafTop5: $\{LID, LCONF\}^5$
    - leafTop5[RID].LCONF *= rootTop5.RCONF
  - sort leafTop5 in DESC
  - Map each LID in leafTop5 to GID using LID_GID.map
  - If test.GID **not in** leafTop5
    - Increment top5_error
- COMPUTE top5_error in %tage

Figure 9. Algorithm to compute top 5 error rates for 10K categories as evaluated by 2-level hierarchical deep learning algorithm

Figure 10. The test image belongs to Wilson's warbler. The predicted categories in order of confidence are a) Yellow Warbler b) Wilsons Warbler c) Yellow Hammer d) Wren, Warbler e) Audubon's Warbler

Figure 11. The test image belongs to fruit Kumquat. The predicted categories in order of confidence are a) Apricot b) Loquat c) Fuyu Persimmon d) Golder Delicious e) Kumquat

Figure 12. The test image belongs to Racing Boat. The top-5 predicted categories are a) Racing Boat b) Racing shell c) Outrigger Canoe d) Gig e) Rowing boat

Figure 13. The test image belongs to category Oak. The top-5 predicted categories are a) Live Oak b) Shade Tree c) Camphor d) Spanish Oak

ACKNOWLEDGEMENTS

We take this opportunity to express gratitude and deep regards to our mentor Dr. Shankar M Venkatesan for his guidance and constant encouragement. Without his support it would not have been possible to materialize this paper.

REFERENCES

[1] Yann LeCun, Koray Kavukvuoglu and Clément Farabet, “Convolutional Networks and Applications in Vision,” in Proc. International Symposium on Circuits and Systems (ISCAS'10), IEEE, 2010
[2] A. Krizhevsky Ilya Sutskever G. Hinton. "ImageNet classification with deep convolutional neural networks", in NIPS, 2012
[3] K. Simonyan, A. Zisserman, “Very Deep Convolutional Networks for Large-Scale Image Recognition”, in arXiv technical report, 2014
[4] H Lee, R Grosse, R Ranganath, AY Ng, "Convolutional deep belief networks for scalable unsupervised learning of hierarchical representations" in Proceedings of the 26th International Conference on Machine Learning, Montreal, Canada, 2009
[5] Jia Deng, Alexander C. Berg, Kai Li, and Li Fei-Fei, “What Does Classifying More Than 10,000 Image Categories Tell Us?” in Computer Vision–ECCV 2010
[6] Sergey Ioffe, Christian Szegedy, "Batch Normalization - Accelerating Deep Network Training by Reducing Internal Covariate Shift", in arXiv:1502.03167v3
[7] J. Deng, W. Dong, R. Socher, L.-J. Li, K. Li, and L. Fei-Fei., “ImageNet: A Large-Scale Hierarchical Image Database”, in CVPR09, 2009.
[8] George E. Dahl, Dong Yu, Li Deng, Alex Acero, “Context-Dependent Pre-Trained Deep Neural Networks for Large-Vocabulary Speech Recognition” in IEEE Transactions on Audio, Speech & Language Processing 20(1): 30-42 (2012)
[9] S. Haykin, "Neural Networks: A Comprehensive Foundation", 3rd ed., New Jersey, Prentice Hall, 1998.
[10] G. Orban, "Higher Order Visual Processing in Macaque Extrastriate Cortex" ,Physiological Reviews, Vol. 88, No. 1, pp. 59-89, Jan. 2008, DOI: 10.1152/physrev.00008.2007.
[11] CCV-A Modern Computer Vision Library, "libccv.org", [Online] 2015, http://libccv.org/
[12] Matthew D. Zeiler and Rob Fergus,”Stochastic Pooling for Regularization of Deep Convolutional Neural Networks”, in ICLR, 2013.

[13] C. R. Dance, J. Willamowski, L. Fan, C. Bray and G. Csurka, "Visual categorization with bags of keypoints", in ECCV International Workshop on Statistical Learning in Computer Vision, 2004.

[14] S. Lazebnik, C. Schmid and J. Ponce, "Beyond bags of features: Spatial pyramid matching for recognizing natural scene categories", in IEEE Computer Society Conf. Computer Vision and Pattern Recognition, 2006.

AUTHORS

Atul Laxman Katole

Atul Laxman Katole has completed his M.E in Jan 2003 in Signal Processing from Indian Institute of Science, Bangalore and is currently working at Samsung R&D Institute India-Bangalore. His technical areas of interest include Artificial Intelligence, Deep Learning, Object Recognition, Speech Recognition, Application Development and Algorithms. He has seeded & established teams in the domain of Deep Learning with applications to Image & Speech Recognition.

Krishna Prasad Yellapragada

Krishna Prasad Yellapragada is currently working as a Technical Lead at Samsung R&D Institute India-Bangalore. His interests include Deep Learning, Machine Learning & Content Centric Networks.

Amish Kumar Bedi

Amish Kumar Bedi has completed his B.Tech in Computer Science from IIT Roorkee, 2014 Batch. He is working in Samsung R&D Institute India- Bangalore' since July 2014. His technical areas of interest include Deep Learning/Machine Learning and Artificial Intelligence.

Sehaj Singh Kalra

Sehaj Sing Kalra has completed his B.Tech in Computer Science from IIT Delhi, batch of 2014. He is working in Samsung R&D Institute India - Bangalore since June 2014. His interest lies in machine learning and its applications in various domains, specifically speech and image.

Mynepalli Siva Chaitanya

Mynepalli Siva Chaitanya has completed his B.Tech in Electrical Engineering from IIT Bombay, batch of 2014. He is working in Samsung R&D Institute India - Bangalore since June 2014. His area of interest includes Neural Networks and Artificial Intelligence.

INTENTIONAL BLANK

SDL BASED VALIDATION OF A NODE MONITORING PROTOCOL

Anandi Giridharan and Pallapa Venkataram*

Indian Institute of Science, Department of ECE, INDIA
anandi@ece.iisc.ernet.in, pallapa@ece.iisc.ernet.in

ABSTRACT

Mobile ad hoc network is a wireless, self-configured, infrastructure less network of mobile nodes. The nodes are highly mobile, which makes the application running on them face network related problems like node failure, link failure, network level disconnection, scarcity of resources, buffer degradation, and intermittent disconnection etc. Node failure and Network fault are need to be monitored continuously by supervising the network status. Node monitoring protocol is crucial, so it is required to test the protocol exhaustively to verify and validate the functionality and accuracy of the designed protocol. This paper presents a validation model for Node Monitoring Protocol using Specification and Description L language (SDL) using both Static Agent (SA) and Mobile Agent (MA). We have verified properties of the Node Monitoring Protocol (NMP) based on the global states with no exits, deadlock states or proper termination states using reach ability graph. Message Sequence Chart (MSC) gives an intuitive understanding of the described system behaviour with varying node density and complex behaviour etc.

KEYWORDS

SDL (Specification and Description Language), validation, verification, Node Monitoring Protocol, safety and liveness property.

1. INTRODUCTION

Node Monitoring is one the important task of fault management in networks, where Mobile Agents have proved that they are very efficient in node monitoring[1]. The usage of Mobile Agents gives the solution to the scalable problem in centralized network management[2]. Mobile Agents plays a vital role in node monitoring process. Agents carry out management function in an autonomous and efficient way[3]. This paper presents a formal model of the Node Monitoring Protocol based on SDL using the Finite State Model. Formal description using SDL specifies the functional operation of the protocol and also helps in detecting design errors like deadlock, livelock, unspecified reception, non-executable interactions, etc. The rest of the paper is organized as follows. Section 2 discusses on Significance of Node Monitoring Protocol (NMP) in Ubiquitous environment. Section 3 presents Formal SDL specification of NMP. Section 4. illustrates validation of NMP for various design errors like deadlock, unspecified reception, livelocks, etc. Section 5 presents Validation results of NMP using reachability graph. Section 6 draws the conclusion.

David C. Wyld et al. (Eds) : CSEN, AISO, NCWC, SIPR - 2015
pp. 95–111, 2015. © CS & IT-CSCP 2015 DOI : 10.5121/csit.2015.51409

2. SIGNIFICANCE OF NODE MONITORING PROTOCOL IN UBIQUITOUS ENVIRONMENT

In a Ubiquitous Network, accurate and efficient monitoring of dynamically changing environment is very important in order to obtain the seamless transparency within mobile devices[4]. Monitoring resource allocation scheme for the Unodes, i.e., nodes running a ubiquitous application in a ubiquitous network is very important to check their Quality of Service. Static and Mobile Agent, based technology can provide a good framework to develop monitoring systems for ubiquitous network environment, since it can do complicated works on behalf of a node independently and transparently[5]. Static Agent sends a request to Mobile Agent to collect raw resource information from the nodes like some of the health conditions like node failure, link failure, misbehaviour of the nodes in the network and to report the monitored results to them. Solution for entering the recovery upon validation is worked out that maintains the health of Node Monitoring Protocol [6].

2.1. Finite State Machine Formalism of Node Monitoring Protocol

An Finite State Machine M, is a 5-tuple A= (I, O, S, T, F) I is the Input, O is the output and S is the states and F is the finite sets. The main system which runs at the central node, where Static Agent is deployed for collection of network status information. The Mobile monitoring system is status monitoring segment, which runs in the migrated Mobile Agents. Figure 1. shows the State transition sequence that illustrates that NMP is capable of delivering data without duplication and in right order. Initially Static Agent which resides in the main segment in idle state then if requests arise, creates Mobile Agent and dispatches sending request M_{req} to monitor the status of the node, initiating the timer. Even if channel loses M_{req}, time out occurs triggering retransmission. and time channel correctly delivers the message. Now Mobile agent sends Request to Node 1 and in case channel loses the Request, Time out occurs and retransmission of the data takes place. Request goes to Node 1 and Mobile Agent monitors the node collects the status of the Node like node failure, link failure, energy level, throughput etc, and delivers to the Static Agent and goes into idle state again. Many important properties of requirement specifications can be checked during requirements capture.First of all, requirements characterizing the total behavior of a system may be expressed in terms of temporal modalities (dynamic requirements) including safety and liveness conditions.

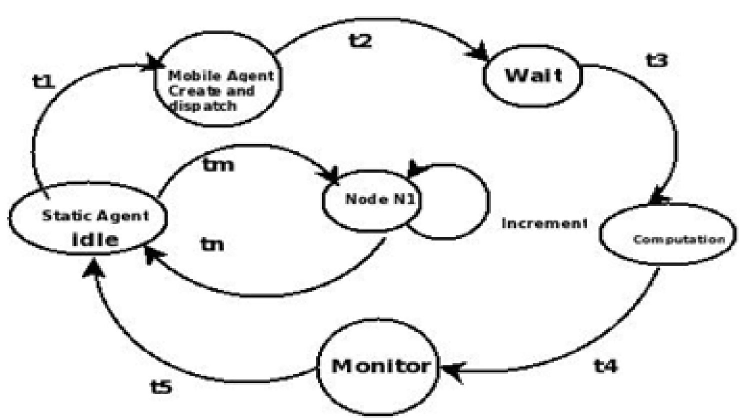


Figure 1: Formal FSM specification of NMP

2.2. NMP Functioning

Liveness property: In system verification and model checking, liveness properties are requirements that something good must eventually happen For example, with every request from Static Agent, Node status should be collected by Mobile Agent and protocol should terminate successfully.

Safety property: Bad things will not happen. For example. Node Monitoring Protocol should operate properly. MSC shows the behavior of the normal Node Monitoring Protocol as shown in the figure 2. We chose to rely on the FSM formalism because it suits very well to the analysis of data flows and allows to put constraints on the variables of the transitions.

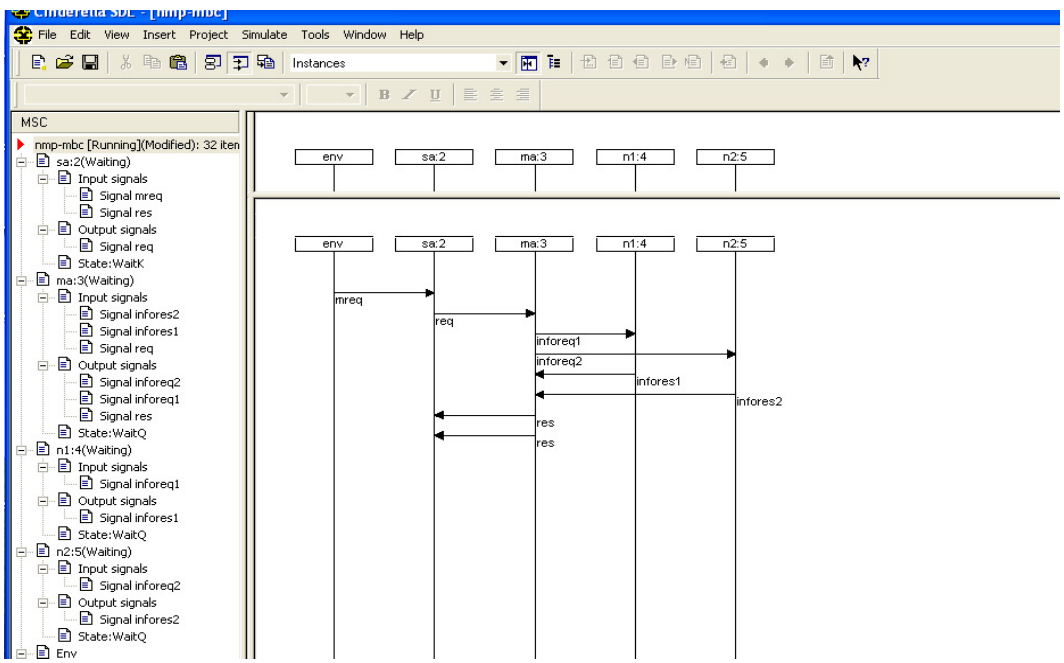


Figure 2: Message Sequence chart showing expected communication between various entities

2.3. Verification of NMP

We manually derived the EFSM directly from the IETF specification. The verification process consists to map the traces of I/O events (messages received and sent) recorded on each node, with the specification. As seen in Figure 3, C_1 is the outgoing channel of the Static Agent and C_2 is the outgoing channel of the Mobile Agent.

Proof of Liveness Property

Liveness property is taken care in design process; they include termination of the protocol. From above transition state, we observe that message M_{req} and Response are transmitted from and to Static Agent respectively even under the conditions of frame and acknowledgement loss and NMP returns to its terminator state. Hence Specified messages have been transmitted and received correctly.

Proof of Safely properties

From transitions, we can see handling of lost frames and Acknowledgement are done by retransmission and no redundancy has occurred by sending two duplicates of the same message. Hence safety property.

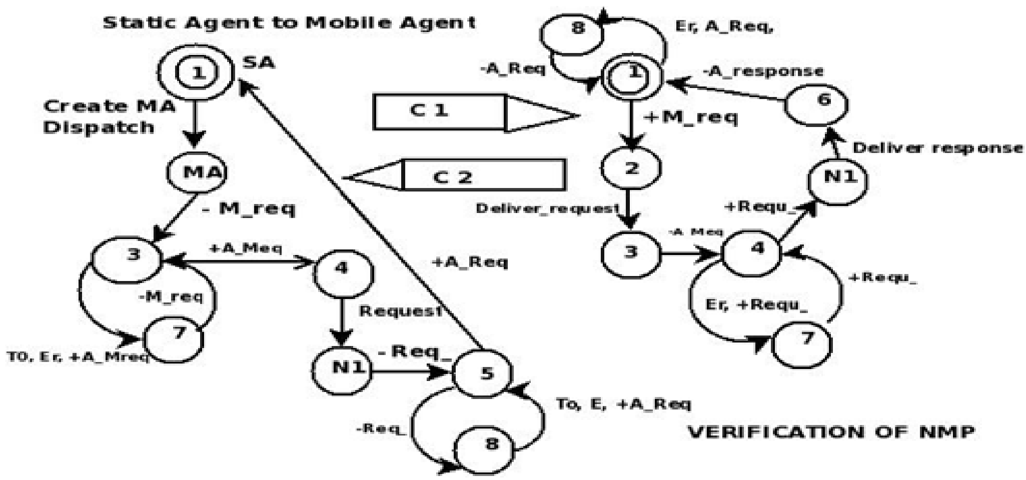


Figure 3: Verification of NMP

3. FORMAL SPECIFICATION OF NODE MONITORING PROTOCOL USING SDL

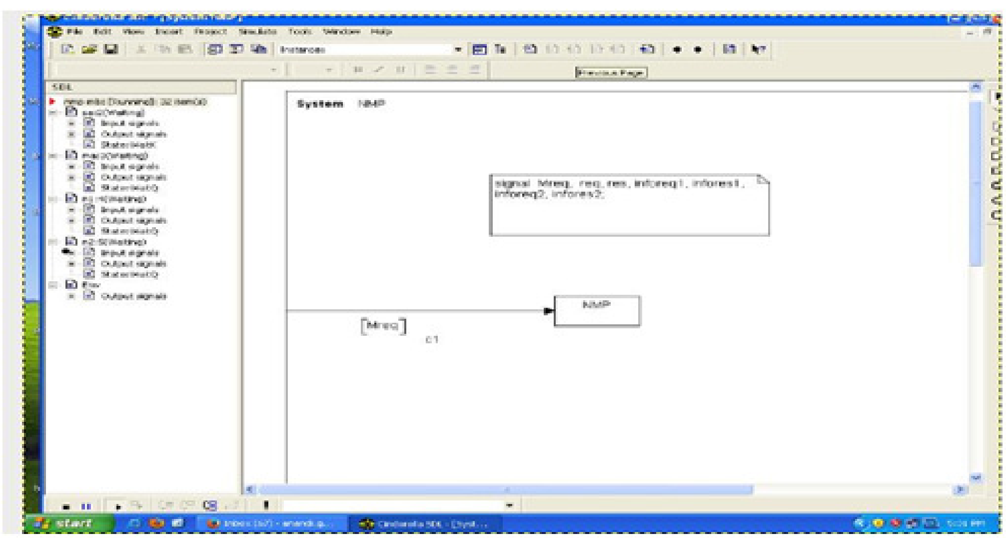


Figure 4: System of Node Monitoring protocol using SDL

We choose SDL (Specification and Description Language) as the target language because it supports more highly-detailed design, so that code automatically generated from the specification can be a much more complete implementation of the system compared to the code generated by

UML. The syntax and semantics of SDL are formally defined, standardized, and maintained by the International Telecommunication Union. Its goal is to specify the behavior of a system from the representation of its functional aspects. The top level of an SDL specification is a system agent consisting of two sub-agents, Static agent and Mobile agent. Blocks of the node monitoring protocol are used to define a system structure as shown in figure 4. We have considered 4 blocks, Static Agent, Mobile Agent, Node 1 and Node 2 as shown in figure 5. Process specifies the behavior of a system from the representation of its functional aspects. We have shown the behavior of the processes of Mobile Agent process, Static Agent process, Node1 process and Node2 process as shown in figures 6, 7, 8 and 9 respectively. Signal routes transfer signal immediately while channels may be delaying. The signal specification identifies the name of the signal type and the sorts of the parameters to be carried by the signal such as Mreq, resp, req, inforequest1, inforesponse1, inforequest2, inforesponse2. As seen in the SDL model, SA behavior is expressed as a process which exists in a state, waiting for an input (event) triggered from environment. When M_{req} signal is sent from environment, SA locates Mobile Agent and sends a request to collect health of the nodes. We have considered two Nodes N1 and N2 in our case. Mobile Agent interacts with the nodes and gets node information back to Static Agent. We have to note that such a specification may contain few errors during its design even from the requirements. For this reason, we have used model checking technique like generating Message sequence chart in order to verify our specification. Indeed, before validating an implementation we need to make sure that the used specification corresponds to the requirements. Simulation was done to verify that specification is free from deadlocks and live-locks within simulated space. Presence of such dead-locks or livelocks reveals that Node monitoring protocol system does not behave as expected that can be monitored using Message sequence chart that is generated after simulation. MSCs are another valuable description technique for visualizing and specifying inter-system, asynchronous component interaction[7]. MSC strength lies in their ability to describe communication between cooperating processes. There are arrows representing messages passed from a sending to a receiving process.

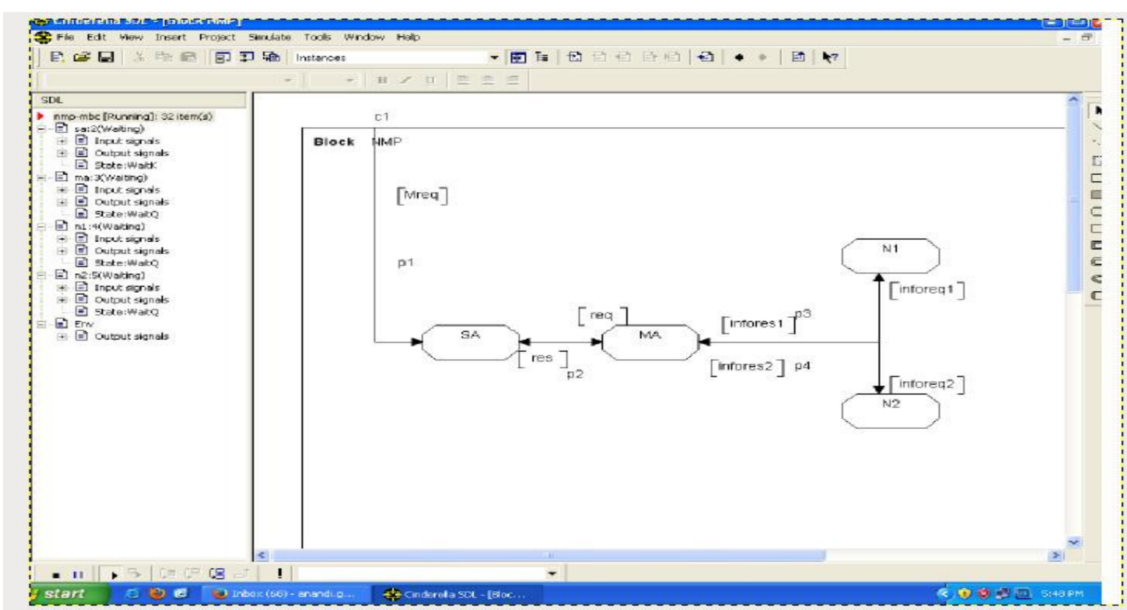


Figure 5: Blocks of Node Monitoring protocol using SDL

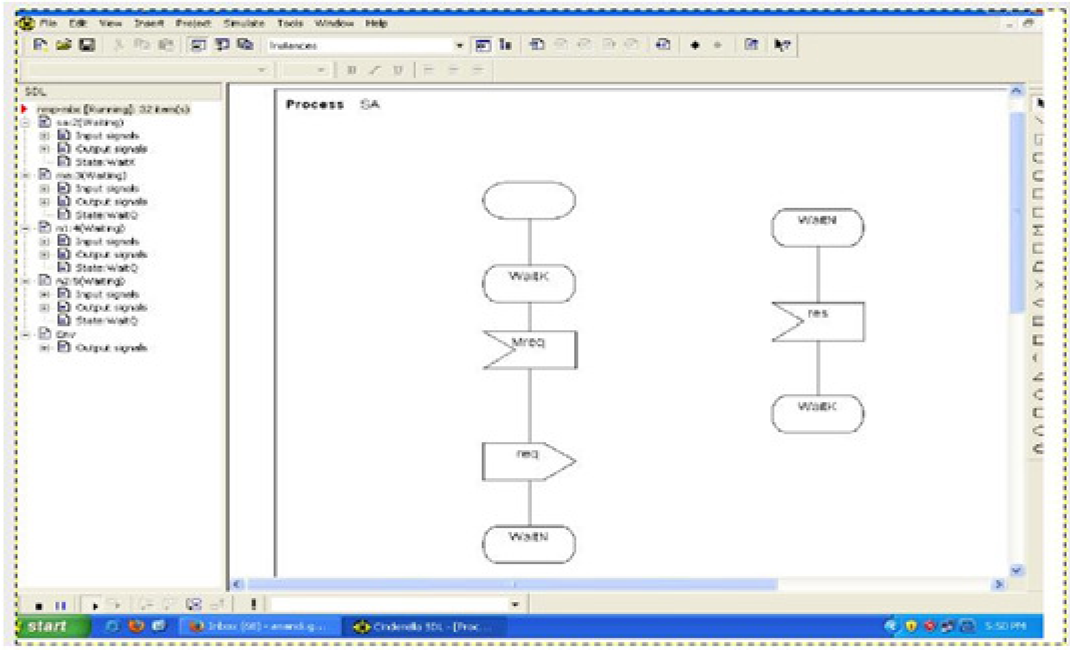


Figure 6: Process Static Agent Of Node Monitoring Protocol Using SDL

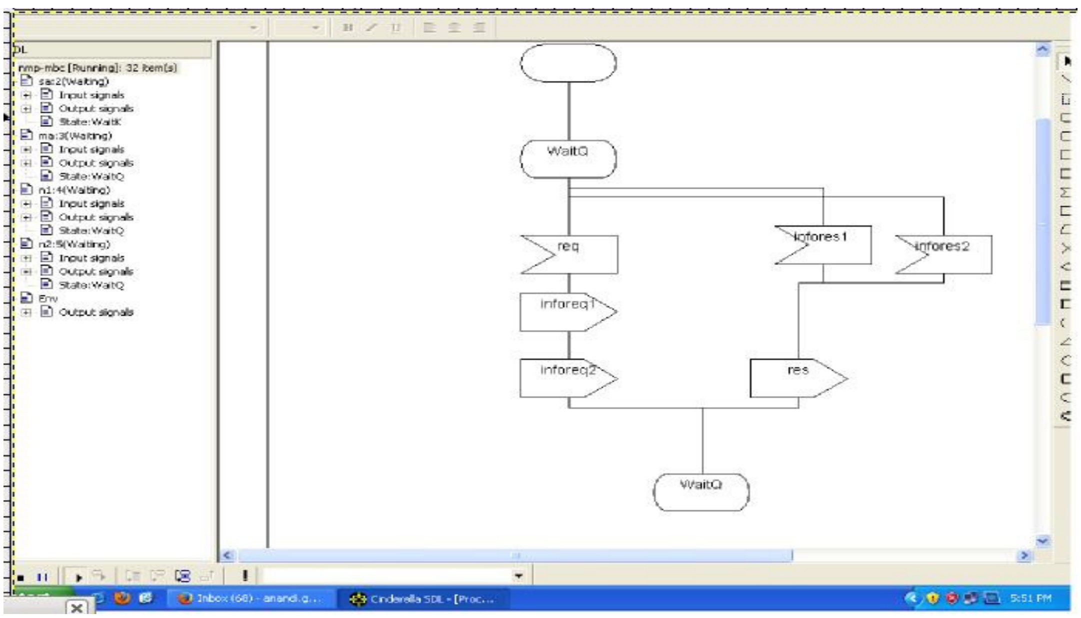


Figure 7: Process Mobile Agent of Node Monitoring Protocol Using SDL

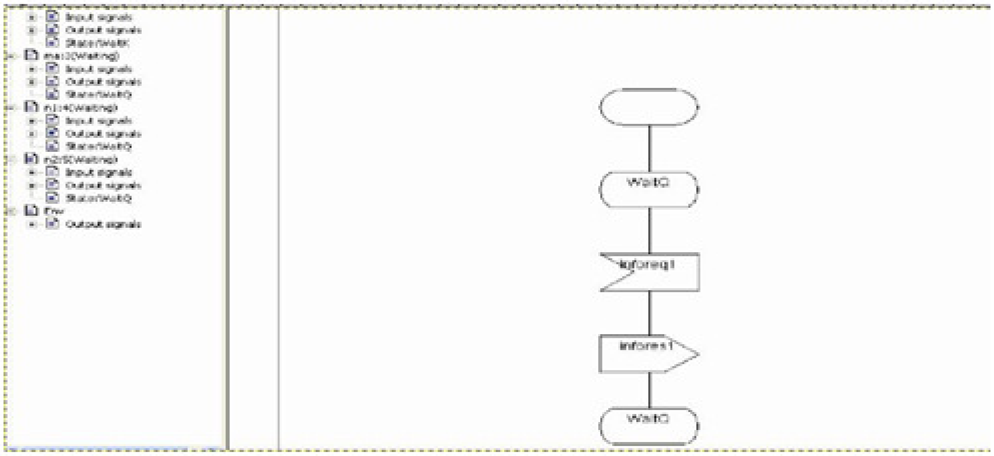


Figure 8: Process of Node 1 of Node Monitoring protocol using SDL

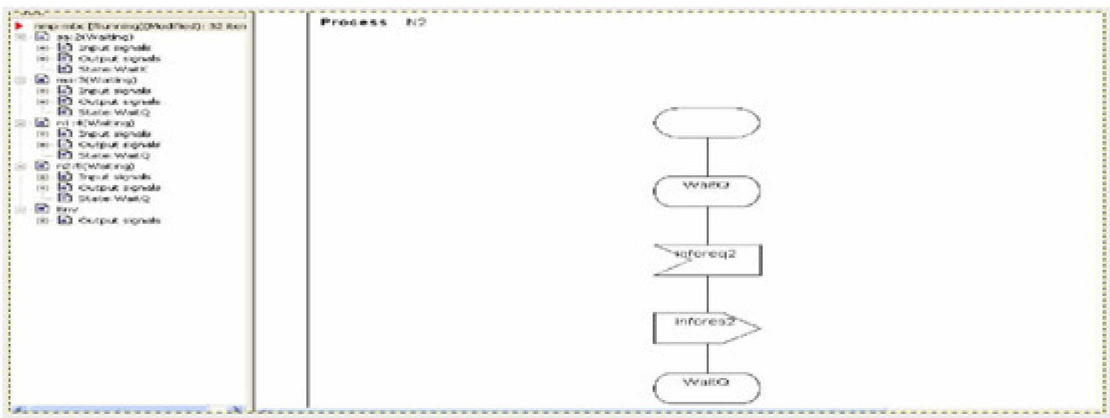


Figure 9: Process of Node 2 of Node Monitoring protocol using SDL

4. VALIDATION OF NODE MONITORING PROTOCOL

Failures may also arise at run-time, for example, because of the loss of network connectivity. node failure, link failure etc. The design of the framework must ensure its ability to hold good under increasing load, increasing complexity of requests and increasing size of resulting composite services[8]. Validation ensures that the protocol specifications will not get into protocol design errors. (Deadlock, unspecified reception, livelock etc). We have used Message sequence charts for validation of Node Monitoring Protocol. MSCs were used to identify different kinds of errors like Deadlock, unreachable states, livelocks etc.

4.1. Deadlock

Deadlock is a scenario, whereby state machines cannot progress to another state because they are waiting for an event that will never occur. Static Agent sends creates Mobile Agent and dispatches, due to the failure of the node, Mobile Agent does not respond to the request of Static Agent. Static Agent waits for random time and time out occurs and again sends request to Static Agent and again goes to wait state. So both the state machines cannot progress further waiting for event to occur that never happens. Hence Deadlock occurs as seen in figure 10**.** Referring to the

Message sequence chart, we can see that Static agent send the request to Mobile Agent. Due to failure of Mobile Agent node, it does not respond. Static agent waits for certain time. Time out occurs and again new request is sent from Static Agent and again goes to wait process expecting Mobile agent to respond, which does not happen. So state machines cannot progress further waiting for event to occur, that never happens. Hence Dead lock occurs. Figure 11 shows the MSC of NMP that indicates Deadlock, where 2 process cannot progress further waiting event to occur.

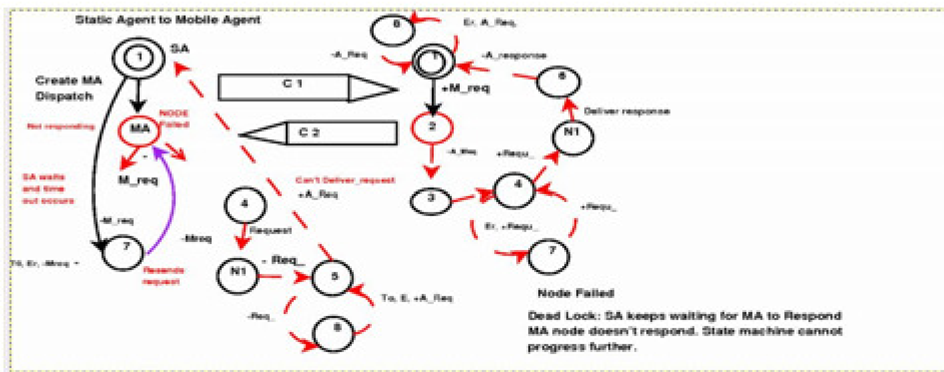


Figure 10: Deadlock error in Node Monitoring protocol

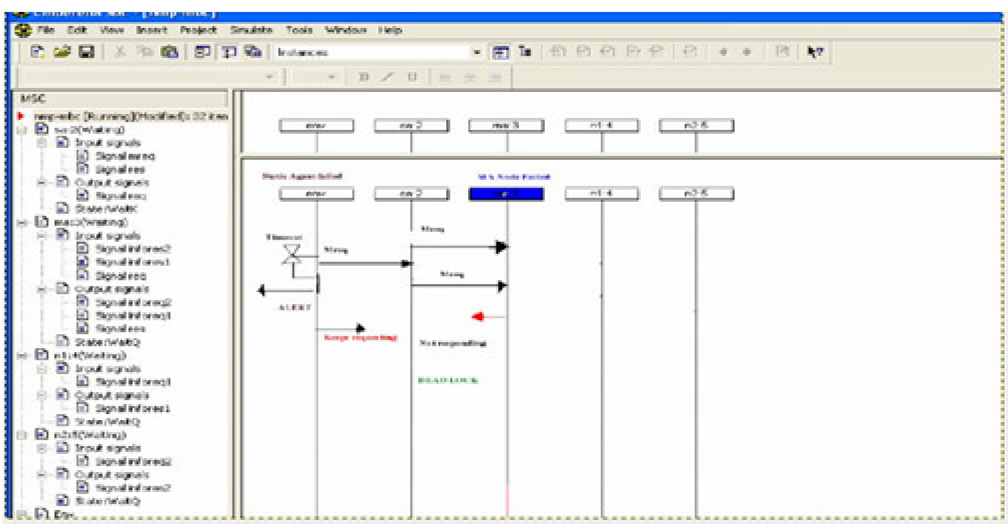


Figure 11: Message Sequence chart showing Dead Lock error in NMP

4.2 Unspecified Reception:

Use of timers may prevent deadlocks, but their use may result in states that are never reached if the specification is faulty[9]. In our simulation, When there was no request from environment, Static Agent is in idle state. Once the request comes from environment , Static Agent sends request to Mobile Agent. Mobile agent goes to Nodes and collects their status. In this case error will propagate because a generic deadlock timer expired that was unaware of the state specific actions to take at this point. So due to ambiguity, Static Agent is not in position to decide what state it should be, hence goes idle. Even through Mobile Agent is ready with node status, Static Agent is not a possible to accept the information as shown in figure 12. Figure 13. shows the

MSC indicating the unexpected state error due to ambiguity. Figure 12: Unexpected State error in NMP.

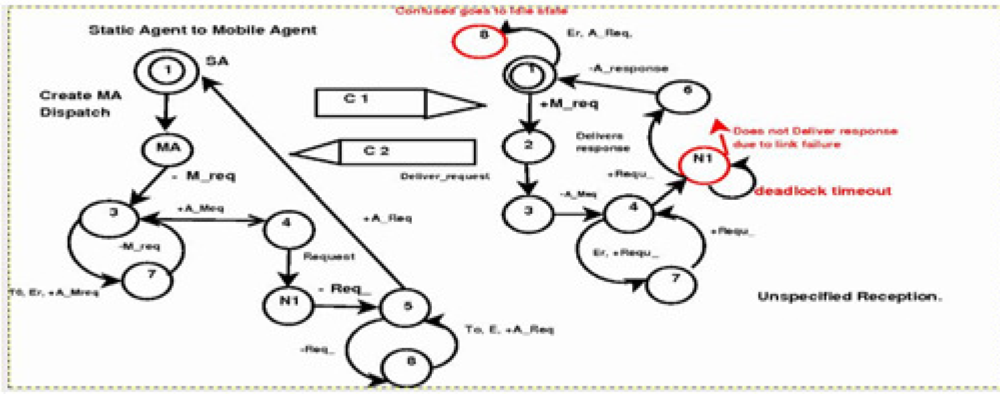


Figure 12: Unexpected State error in NMP

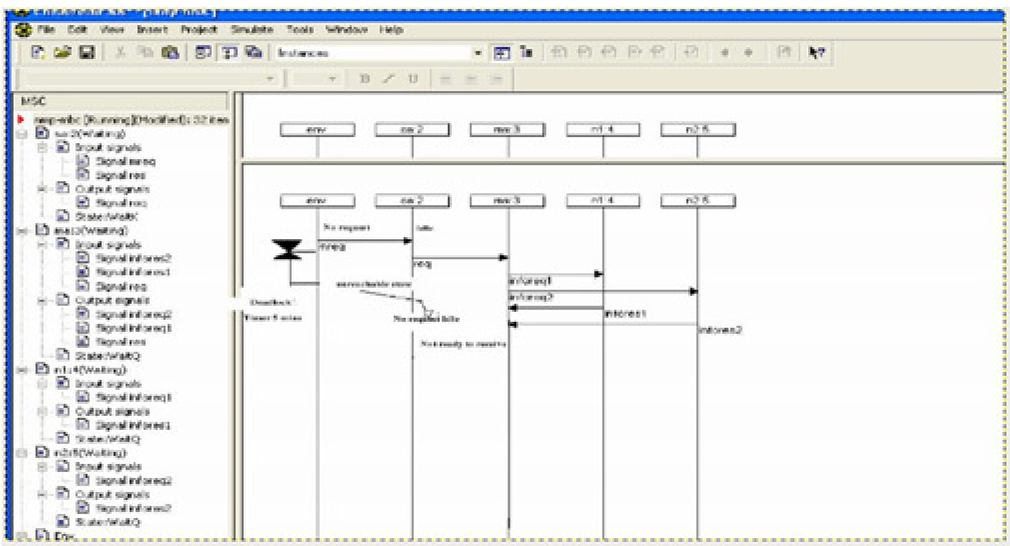


Figure 13: Message Sequence chart showing Unexpected State error in NMP

4.3. Data loss:

As indicated in figure 14, request from Node1 gets lost in channel and no response from Mobile Agent regarding status of the Node. Figure 15. shows that Request sent by Static Agent to Mobile Agent and request gets lost in the channel, Response comes from only from Node2 to Mobile Agent. Data loss occurs, when one or more packets of data travelling across a network fail to reach their destination. Data loss can be caused by a number of factors, including packet drop because of channel congestion, rejected corrupted packets, faulty networking hardware. As seen in the figure 16, it shows that the data loss increases if more number of packets are sent. Hence throughput will be less due to the number of retransmission.

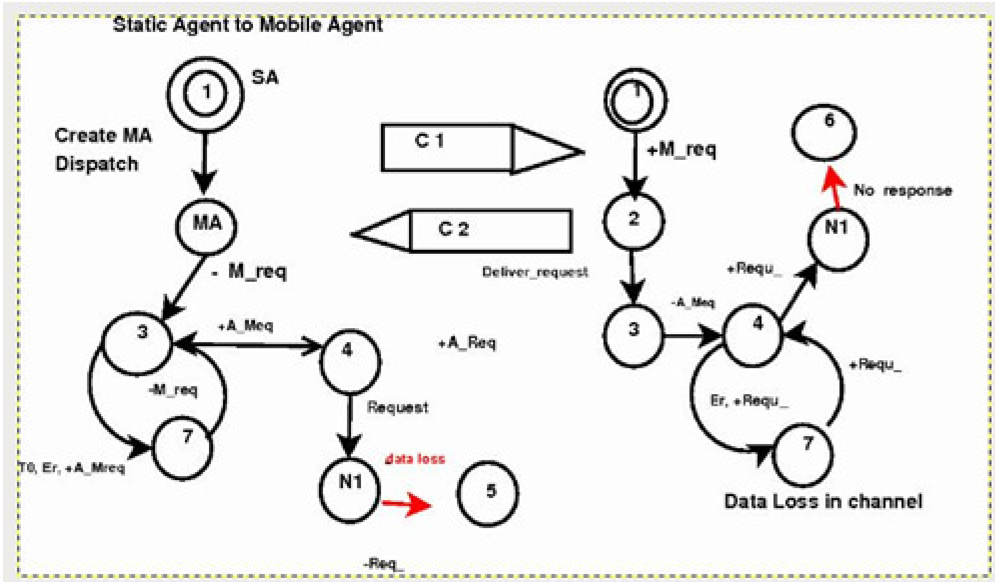


Figure 14: Data Loss occurring in channel

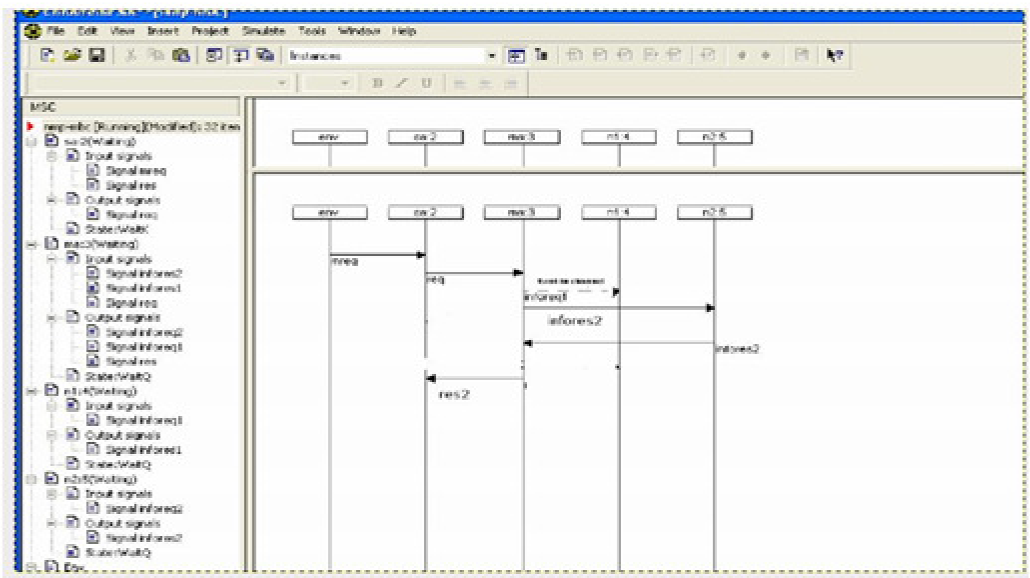


Figure 15: Message Sequence chart showing Data Loss occurring in channel

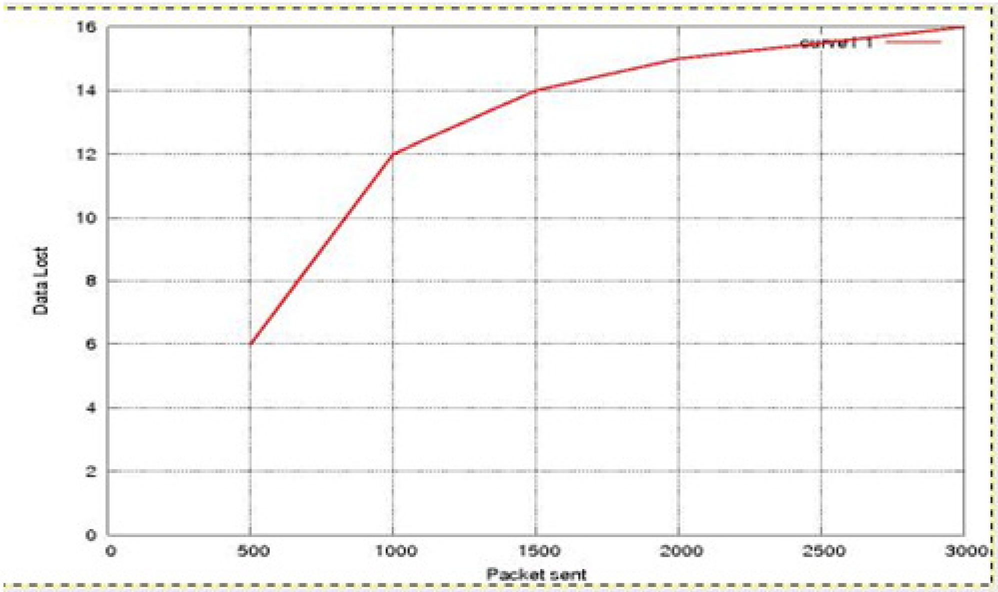


Figure 16: Data loss verses number of packets sent

4.4 Livelocks:

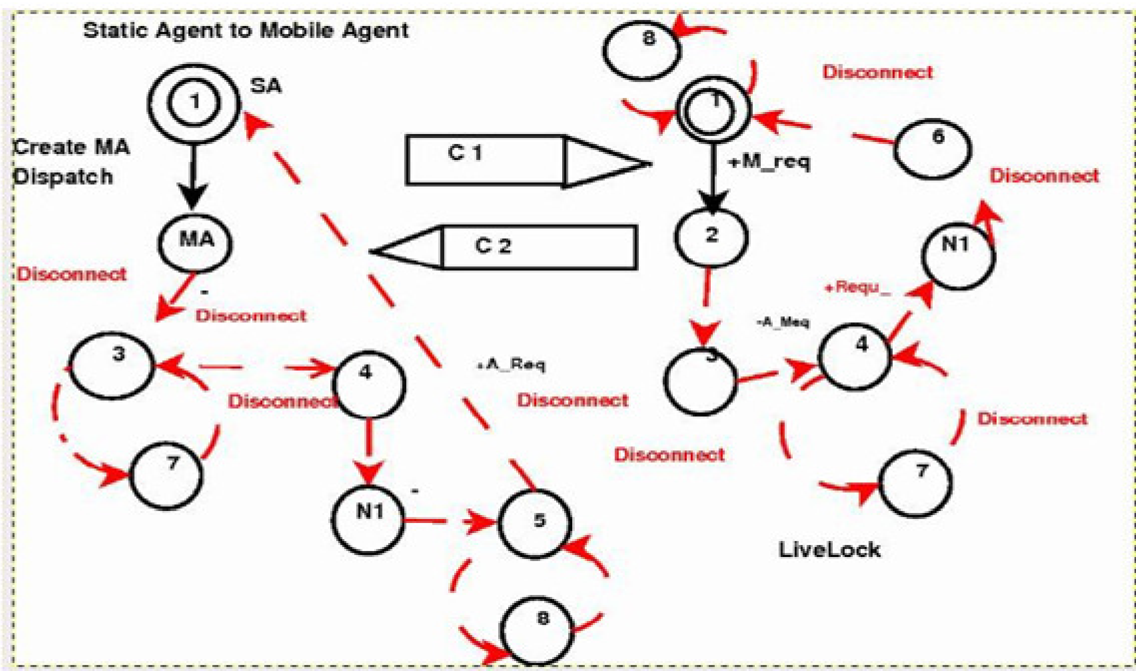


Figure 17: Message Sequence chart shows infinite loop livelock error

Livelock is a scenario whereby sequences of messages is repeated in an endless loop as shown in figure 17. Without appropriate safety mechanisms livelock can consume all of the resources in a network. Livelocks can occur depending on the value of data, such as an entity forwarding a message to itself . MSC indicates, how sequence of messages are repeated in an endless manner as shown in figure 18.

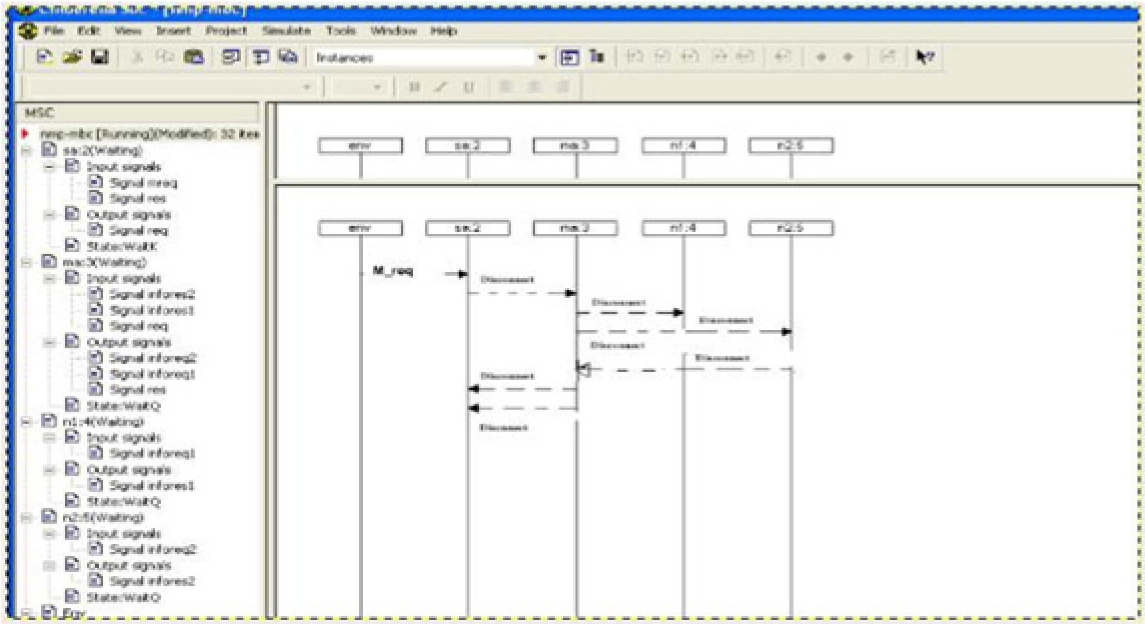


Figure 18: Message Sequence chart shows infinite loop livelock error

5. VALIDATION RESULTS OF NMP USING REACHABILITY GRAPH

The most straightforward technique to validate a given network of two communicating FSMs is called state exploration. We have considered Node monitoring Network [Mobile Agent, Node] whose communicating FSMS sender machine and reaching machine are as shown in figure 19.. The exchanged messages between two machines have the following meaning:

-M $_{req}$ denotes a request sent to Mobile Agent from environment.
+M req denotes Postive acceptance of M req.
-Req denotes a request sent from Mobile Agent to Node.
+Req denotes postive acceptance of request from Node.
-A req Acknowledgement sent from Mobile Agent.
+A req Positive Acknowledgement from Node to Mobile Agent.

In order to describe the behavior of our network, many processes have been specified and tested. Specification may contain errors like deadlock, unspecified reception, data loss etc. For this reason, using reachablility analysis, we had to validate our specification. During validating FSM, we verified that specification had errors like deadlock, unspecified reception and one process terminated successfully.

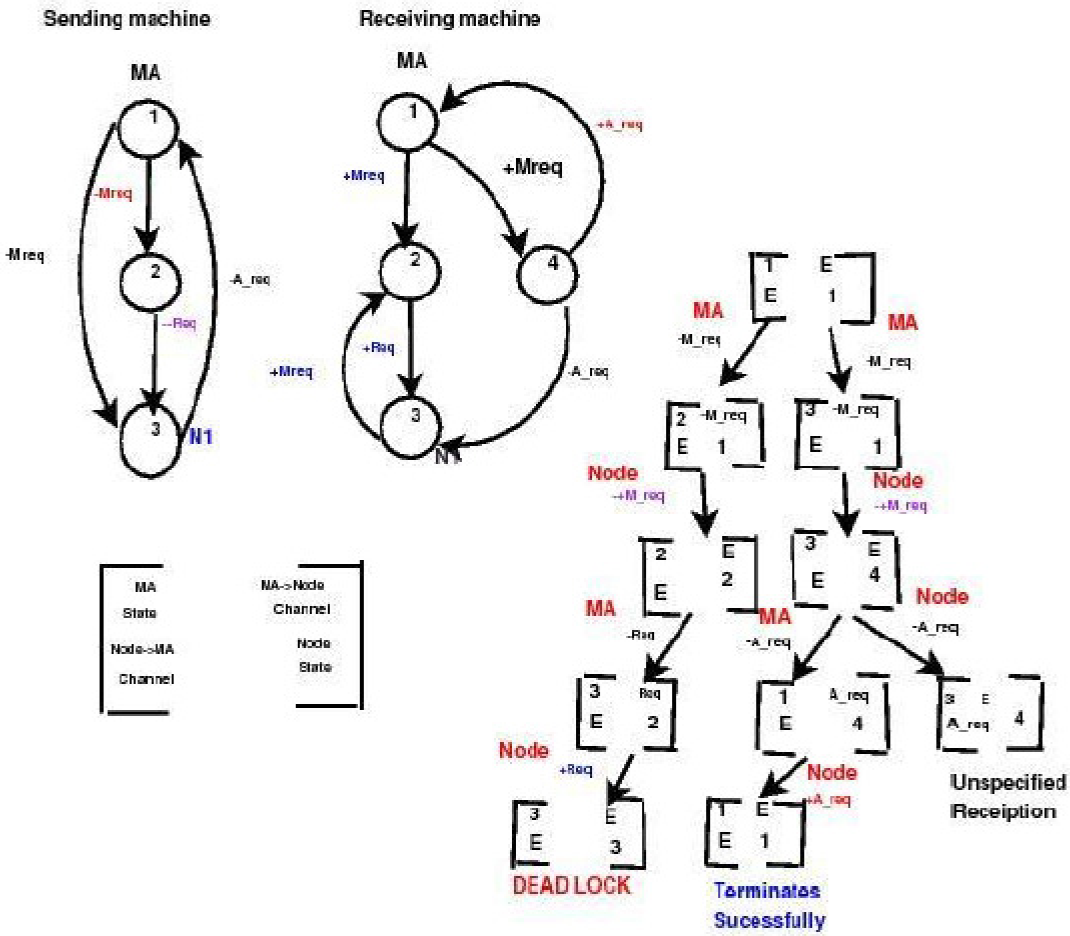


Figure 19: Reachability Graph for Node Monitoring Protocol

6. SIMULATION AND RESULTS

Simulation was used in both protocol specification and validation using Cinderella SDL tool to conduct verification and validation of Node Monitoring Protocol. We simulated on five to fifteen nodes. It was found that various errors increases as the traffic on the network increased. Simulation results on data loss, deadlock error, unspecified error and performance of the protocol are summarized below.

6.1 Dead Lock

Deadlocks occur when two or more processes interfere with each other in such a way that the network as a whole eventually cannot proceed. Multiple processes, and multiple processes have always given rise to deadlocks of various kinds. Graph 20. shows Dead-lock error rate versus Number of processes. As the number of process increased, the deadlock error also increases.

6.2 Unspecified Error

Simulation was conduction on cinderella SDL tool for 5, 10 and 15 nodes, we found that as the unspecified error increases delay increases as seen in the figure 21 and also we found that as the number of nodes increased unspecified reception error also increased as seen in figure 22.

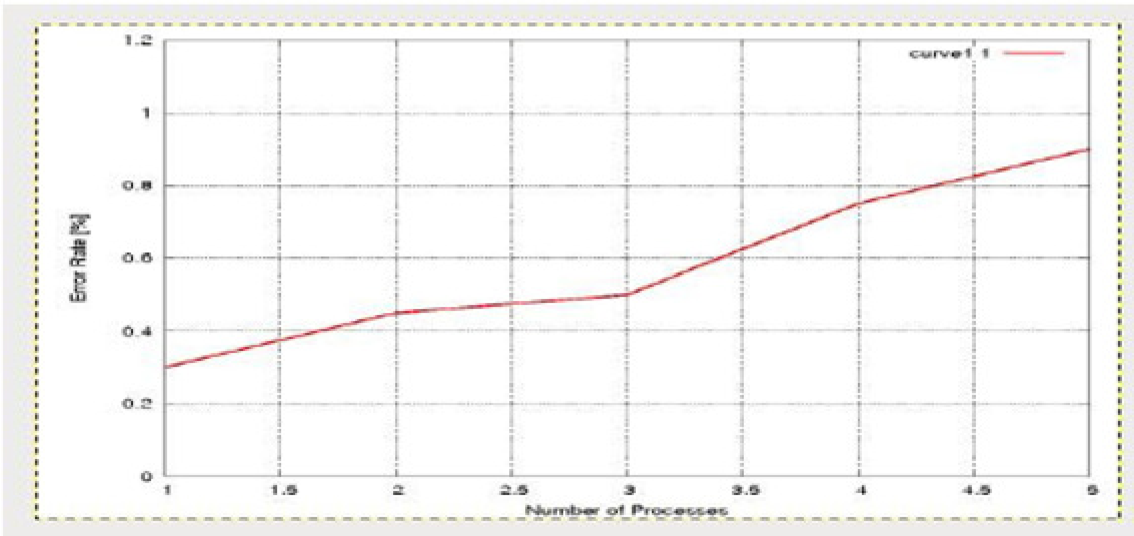


Figure 20: Deadlock error verses number of process

6.3 NMP performance

NMP performance is an overall measure of the effciency of the system's achievement in terms of rates and throughtputs. The results of applying variation in data transmissions versus Error rate are drawn in figure 23. with two sets of settings. Curve 1 and 2 were for heavy data transmission rate and slower data transmission rate. The larger data transmission rate, more the error rate. This is due to messages are lost in the network. On the other hand, decreasing date rate below certain value will discard reply messages that may arrive a bit later. The best choice of data transmission is that one with less errors as shown in figure 24 . It is obvious that performance improves only when the bottleneck transition time is decreased.

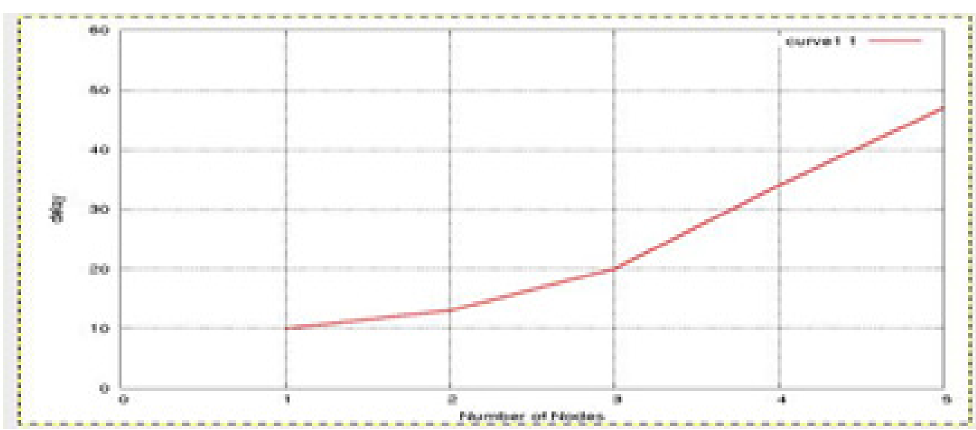


Figure 21: Delay verses number of Nodes

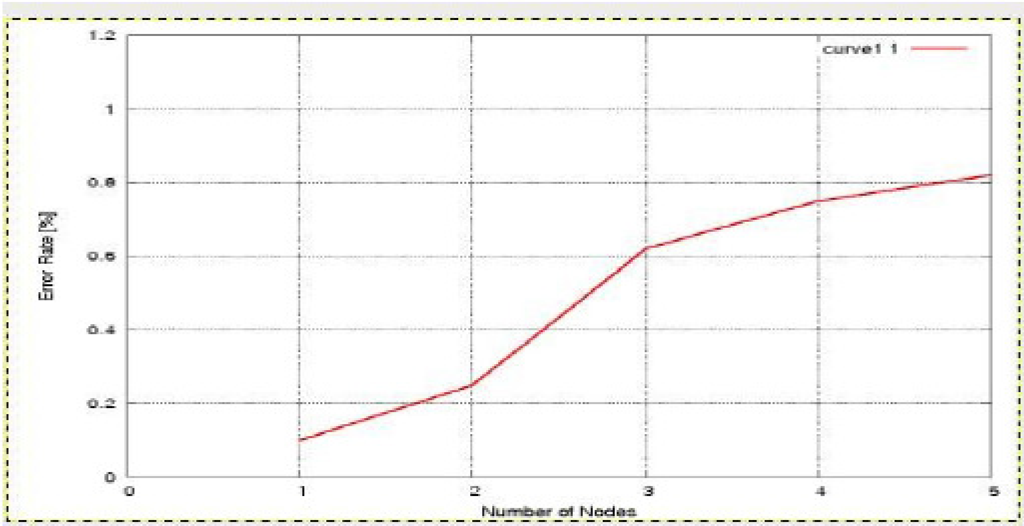


Figure 22: Deadlock error vs number of Nodes

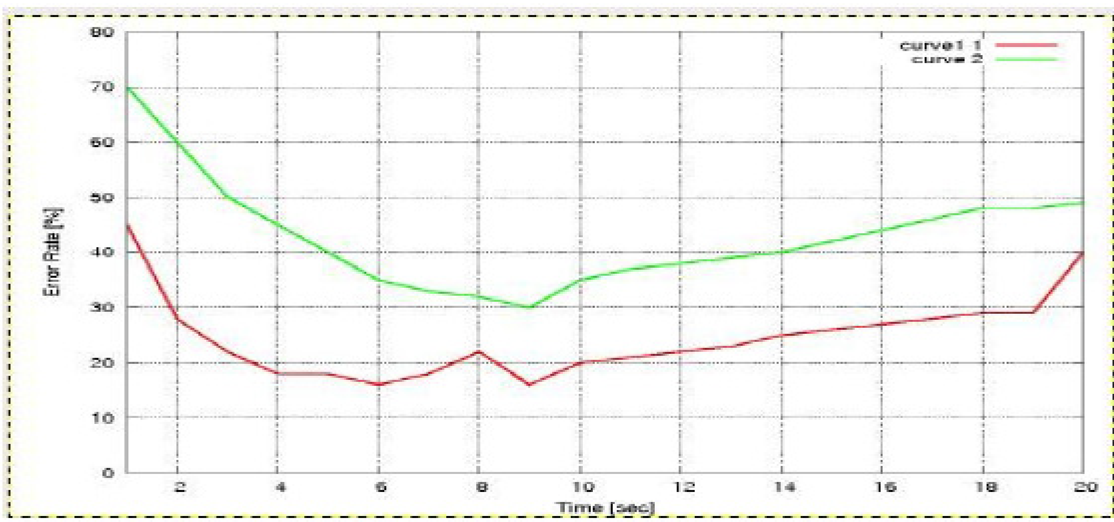


Figure 23: High Data transmission rates versus Error Rate

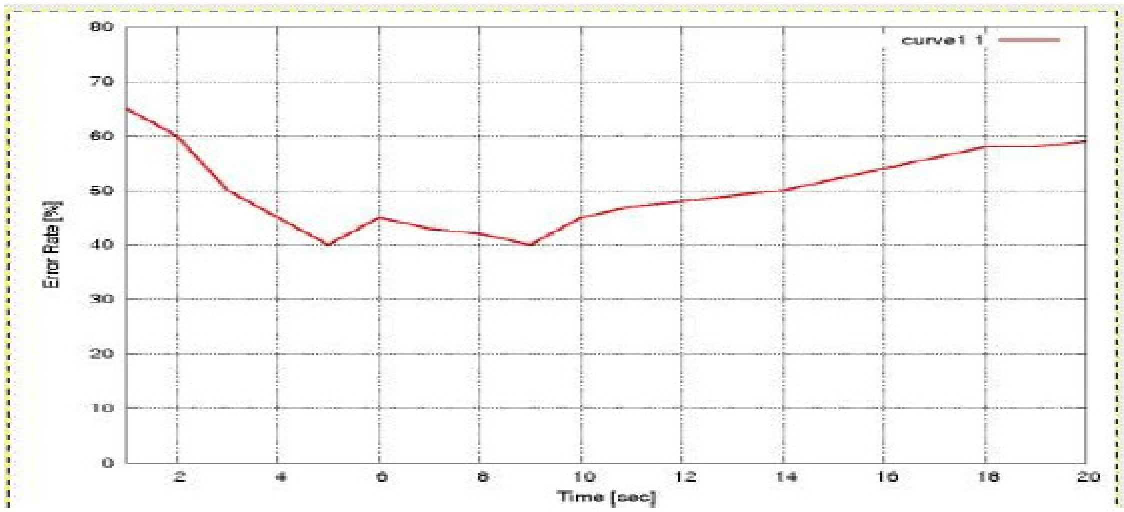


Figure 24 : Light Data Transmission Rate Versus Error Rate

7. CONCLUSION

This paper has presented verification and validation model for the Node Monitoring protocol. It includes a formal specification of the protocol using Specification and Description Language and Message sequence charts a method and a tool for the automated test generation of scenarios. Validation checks for safety and liveness property of the protocol to check proper functioning and termination of protocol and validation model presents several advantages[10][11]. Reachability analysis was carried out to check the correctness properties of NMP. First, the design of a formal specification from which tests are generated contributes to eliminate design errors like Deadlock, unspecified receptions and livelocks and using SDL, it is shown that design flaws and ambiguity introduced in informally specified, textual protocols can be avoided if protocol is formally modelled.

REFERENCES

[1] Tatjana Kapus. “Specifying System Families with TLA+”, Recent Researches in Engineering Education and Software Engineering, pp. 98-103, (2012).

[2] Sarada Prasad Gochhayat and Pallapa Venkataram, Performance Analysis of Node Monitoring Protocol in Ubiquitious Networks, Ubiquitous Computing and Communication Journal,vol 8, issue 1, May 2013.

[3] Richard Castanet, Marcien MacKaya, Patrice Laurenscot , A Multi-service and Multi-protocol Validation Platform Experimentation Results, Testing of Communicating Systems Lecture Notes in Computer Science Volume 2978, 2004, pp 17-32.

[4] C. Bohoris, G. Pavlou, and H. Cruick- shank, Using mobile agents for network performance management, in Network Operations and Management Sympo- sium, IEEE/IFIP,2000,pp. 637652.

[5] H. H. To, S. Krishnaswamy, and B. Srinivasan, Mobile agents for network management when and when not,in Proceedings of the 2005 ACM symposium on Applied computing, SAC 05. NewYork, NY, USA: ACM,2005, pp. 4753.

[6] M. Konaand C. Z. Xu, A framework for network management using mobile agents, in Proceedings of the 16th International Parallel all optical networks, Network and Service and Distributed Processing Symposium, ser. IPDPS'02. Wash-ington, DC, USA, IEEE Computer Society, 2002, pp. 714.

[7] D. Gavalas, G. E. Tsekouras, and C. Anagnos- topoulos, A mobile agent plat- form for disttributed network and systems management, J. Syst. Softw., vol. 82, no. 2, Feb. 2009, pp.355371.

[8] D. Gavalas, D. Greenwood, M. Ghanbari, and M.O Mahony, Using mobile agents for distributed network performance management,in 3rd International Workshop on Intelligent Agents for Telecommunication Applications, 1999.

[9] Channappagoudar, Mallikarjun B, Pallapa Venkataram, Mobile agent based node monitoring protocol for MANETs, World Academy of Science, Engineer- ing and Technology. International Journal of Computer, Information, Systems and Control Engineering Vol:8 No:1, 2014,vol 8, No:1, 2014.

[10] Ameer A. Abbasi, Mohamed F. Younis, Senior Member. IEEE, and Uthman A. Baroudi , Recovering from a node failure in wireless Sensor, Actor Network with Minimal Topology changes , IEEE Transactions on vehicular technology, vol.62, no.1 Jan 2013.

[11] G. Wang et al., Sensor relocation in mobile sen sor networks, in proc. 24 th Annu. Joint Conf. INFOCOM, Miami, FL, Mar. 2005,pp.2302- 2312.

[12] Group Context-aware Service Discovery for Supporting Continuous Service Availability, Saehoon Kang, Woohyun Kim, Dongman Lee, and Younghee Lee.

AUTHORS

Anandi Giridharan, received MSc(Engg) from Indian Institute of Science. She currently working as Senior Scientific Officer in ECE Department, Indian Institute Science, Bangalore. Her Research Interest are in area of Ubiquitous Learning, Communication Protocols and Multimedia systems.

Prof. Venkataram Pallapa received his Ph.D. Degree in Information Sciences from the University of Sheffield, England, in 1986. He is currently the chairman for center for continuing education, and also a Professor in the Department of Electrical Communication Engineering, Indian Institute of Science, Bangalore, India. Dr. Pallapa's research interests are in the areas of Wireless Ubiquitous Networks, Communication Protocols, Computation Intelligence applications in Communication Networks and Multimedia Systems. Dr. Pallapa is the holder of a Distinguished Visitor Diploma from the Orrego University, Trujillo, PERU. He has published over 150 papers in International/national Journals/conferences. Written three books: Mobile and wireless application security, Tata McGraw-Hill, Communication Protocol Engineering, publications Prentice-Hall of India (PHI), New Delhi, 2014 (Co-author: Sunil Manvi, B Satish Babu) and Multimedia: Concepts & Communication, Darling Kinderley(India) Pvt. Ltd., licensees of Pearson Education in South Asia, 2006. Written chapters for two different books, and a guest editor to the IISc Journal for a special issue on Multimedia Wireless Networks. He has received best paper awards at GLOBECOM'93 and INM'95 and also CDIL (Communication Devices India Ltd) for a paper published in IETE Journal. He is a Fellow of IEE (England), Fellow of IETE(India) and a Senior member of IEEE Computer Society.

INTENTIONAL BLANK

A COMPARATIVE ANALYSIS OF RETRIEVAL TECHNIQUES IN CONTENT BASED IMAGE RETRIEVAL

Mohini. P. Sardey[1] and G. K. Kharate[2]

[1]AISSMS Institute Of Information Technology, Savitribai Phule Pune University, Pune
(Maharashtra State), India.
sardeymp@yahoo.com
[2]Matoshri COE and Research Center, Savitribai Phule Pune University, Pune
(Maharashtra State), India.
gkkharate@gmail.com

ABSTRACT

Basic group of visual techniques such as color, shape, texture are used in Content Based Image Retrievals (CBIR) to retrieve query image or sub region of image to find similar images in image database. To improve query result, relevance feedback is used many times in CBIR to help user to express their preference and improve query results. In this paper, a new approach for image retrieval is proposed which is based on the features such as Color Histogram, Eigen Values and Match Point. Images from various types of database are first identified by using edge detection techniques .Once the image is identified, then the image is searched in the particular database, then all related images are displayed. This will save the retrieval time. Further to retrieve the precise query image, any of the three techniques are used and comparison is done w.r.t. average retrieval time. Eigen value technique found to be the best as compared with other two techniques.

KEYWORDS

Edge detection, Match Points, Eigen Values, Histogram

1. INTRODUCTION

Key task of the computer science is always been the management of digital information. Years ago the data used to be in terms of numbers and [1] [4] [9] text; relational databases handled the storage and process of searching. But with the rapid growth of complex data types e. g. images, sound, videos, there is a need to research that suit our changing needs better. So in last two decades, a new approach data management has been studied extensively to address these requirements. In variety of application areas based on content based or on similarity searching has become fundamental computational task [18] which includes multimedia information retrieval, data mining, pattern recognition, data compression, biomedical databases, statistical data analysis. Rest of the paper is organized as follows. Section 2 illustrates frame work of CBIR,

David C. Wyld et al. (Eds) : CSEN, AISO, NCWC, SIPR - 2015
pp. 113–121, 2015. © CS & IT-CSCP 2015 DOI : 10.5121/csit.2015.51410

section 3 explains various techniques used to implement the objective, section 4 will tabulate the result and finally section 5 draws the conclusion remark and future scope is discuss.

2. CBIR FRAMEWORK

Retrieval system proposed in this paper is described by the frame work or the block diagram as depicted in figure 1. This block diagram explains the flow of the proposed technique very easily which every reader can understand.

Image retrieval system can be conceptually described by the framework depicted in figure 1. In this article we survey how the user can formulate a query, which is the appropriate retrieval technique for various types of image database such as Face, Vehicle, Animal and Flower, how the matching can be done [4].

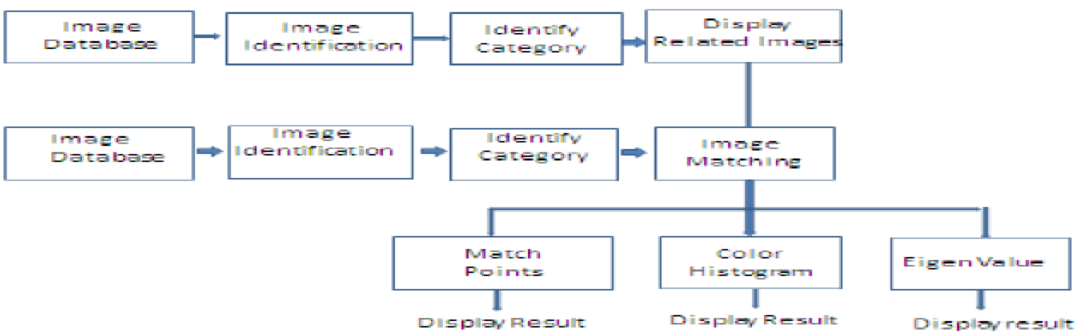


Figure 1. Basic block diagram of CBIR

This paper proposes a technique of image retrieval which first identifies the type of image by using edge detection technique, as shown in fig.2. This step is essential when the images are from different format. When the type of query image is known, then system will search the query image in that particular data type only which will save search time substantially. Query image will be searched precisely by using either of the three image retrieval techniques, Color Histogram, Eigen values base and Match point base.

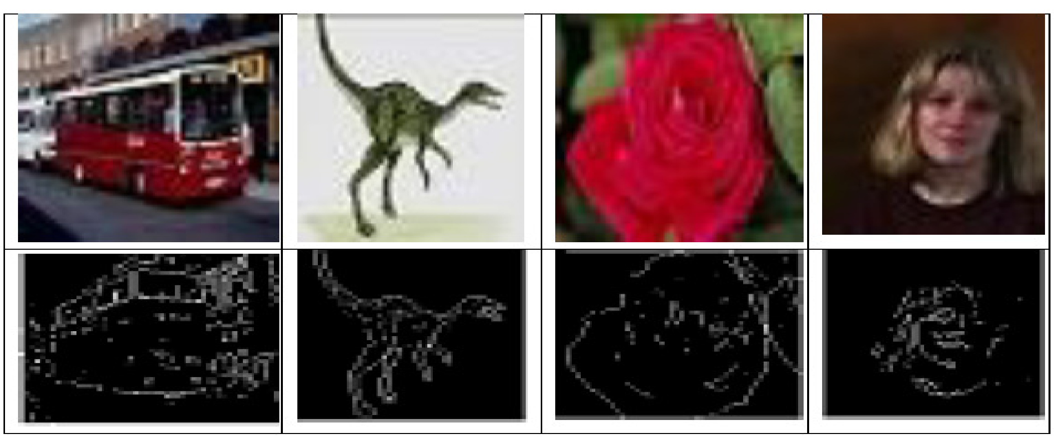


Figure 2. Image and its Edge

3. IMPLEMENTATION TECHNIQUES

3.1 Eigen Values:

The term “eigenvalue” is a partial translation of the German “Eigen wert”. A complete translation would be something like “own value” or “characteristic value,” but these are rarely used. Eigenvalues play an important role in situations where the matrix is a transformation from one vector space onto itself.

When matrix transformation is from one vector space, role played by Eigen value is very important. When applications are based on image processing, eigen value approach plays a prominent role, e.g. measurement of sharpness of an image or segregation of images into images of vehicles or animals, etc. , Aim is to implement the mode with some real time variation, to precise face or image and retrieve it from a large number of stored faces. The Eigen face approach uses Principal Component Analysis (PCA) algorithm for the recognition of the images. It gives us efficient way to find the lower dimensional space.

3.1.1 Sensitivity and accuracy of Eigen value:

Basically, eigen value is a matrix which is susceptible to any deviation or changes i.e. disorder in matrix element will result in significant changes in eigen values. When the operations are related to floating point arithmetic, computations will result in introduction of round-off errors and also have similar effect to the perturbations taking place in original matrix [9]. This will in turn result in the magnification of round off errors in the eigen values that are computed.

Assuming A has full set of linearly independent eigenvectors and using the eigen value decomposition we can get a rough idea of the sensitivity.

Equation of eigen value and eigen vector for a square matrix can be written as

$$(A-\lambda I)\ x = 0,\ x \neq 0$$

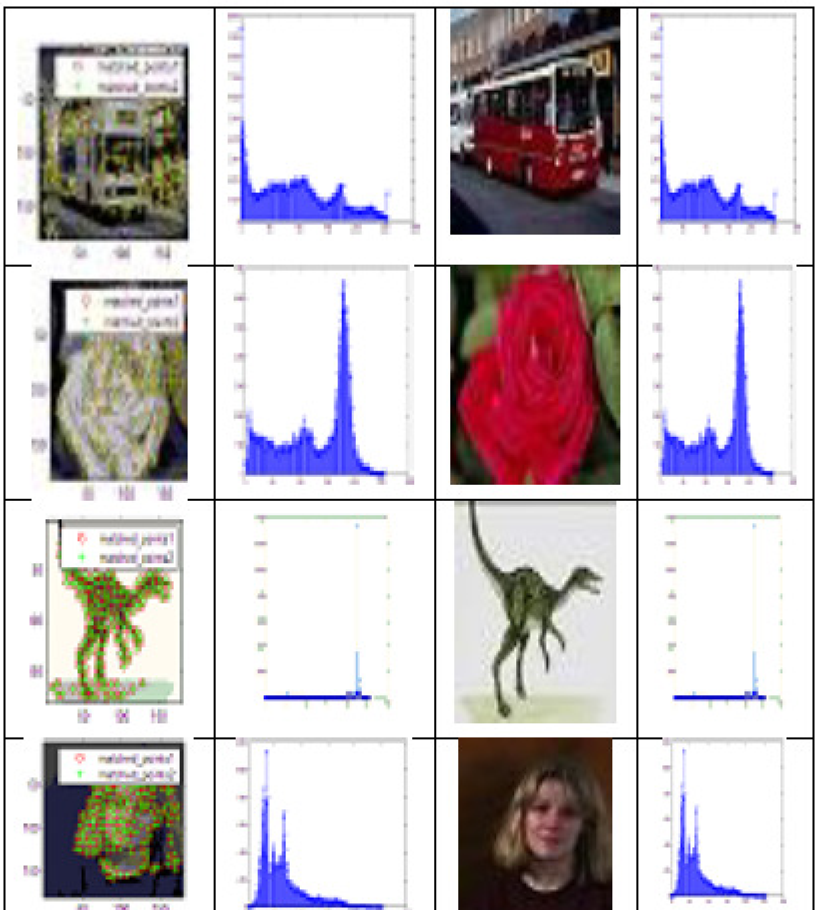
This implies that $(A-\lambda I)$ is singular and hence

$$\det(A.\ \ddot{e}I) = 0$$

This particular definition of eigen value, which excludes the corresponding eigen vector [10], is the characteristic polynomial of A or the characteristic equation and the degree of this polynomial is the order of matrix. Therefore if there are n-eigenvalues, matrix is of size n-by-n.

Table 1. Retrieval time in each image category

Techniques / Database	Match Point (sec)	Histogram	Eigen values
Flower	7.701922(184x2)	12.703803	3.121432
Face	6.544435(187x2)	11.324701	3.008408
Vehicle	6.083464(188x2)	12.393825	3.005223
Animal	5.851284(185x2)	11.062617	3.678530



Results of Match Point	Histogram of image from database	Query image	Histogram of Query image

Figure 3. Results of images in each category

3.2 Color

Color is an important visual attribute for both human perception and computer vision and one of the most widely used visual features in image retrieval [l]. But an appropriate color space and color quantization must be specified along with a histogram representation of an image for retrieval purpose. Histogram describes the global distribution of pixels of an image [17][18]. Main advantage of a color histogram is its small sensitivity to variations in scale, rotation and translation of an image. We utilize different kinds of quantization schemes for the implementation of the color histograms in HSV color space. We observed that the HSV color model is better than the RGB color model for our approach using the following quantization scheme where each color component is uniformly quantized. . Although the color-based methods perform surprisingly well, [14] [15] their performance is still limited to less than 50% in precision. The main reason is because the color representation is low-level, even with the use of pseudo object models.

In general, color is one of the most dominant and distinguishable low-level visual features in describing image. Many CBIR systems employ color to retrieve images, such as QBIC system and Visual SEEK.

The retrieval method of using color characteristic was originally proposed by Swain and Ballard, they put forward the color histogram [6] method of which the core idea is to use a certain color space quantization method for color quantization, and then do statistics for the proportion of each quantitative channel in the whole image color. Abscissa represents the normalized color value, ordinate represents the sum of image pixels which corresponding to each color range [8][9][10]. Image statistical histogram is a one-dimensional discrete function:

$$h_k = n/n_k\,,\ \ k=0,1,\ldots,L-1$$

The letter k presents eigenvalues of color,letter l presents the number of features of value . So we get the color histogram of the image P as follows:

$$H(p)=[h_1,\ \ h_2,\ \ \ldots h_{L-1}]$$

There are many color histogram methods such as the global color histogram, cumulative histogram and sub-block histogram. However, color histogram has its own drawbacks, such as the color histograms of different images may be the same.

3.3 Match Point Based: Computer Vision System Toolbox is used for this feature:

This paper uses the functions from Computer Vision System toolbox to detect the objects using the Viola-Jones algorithm. Detection of corners in a grey scale image another function is used. Another function is used to detect the corners in a grey scale image. It returns location as a matrix of [x, y] coordinates. The object finds corner in an image using Harris corner detection, minimum Eigenvalues or local intensity comparison method. Using another function, feature vectors are extracted from intensity or binary image. These vectors are also known as descriptors and are derived from pixels surrounding an interest point by the function. These pixels match features and describe them by a single-point location specification. The function extracts feature vectors from an input intensity or binary image. These feature vectors, also known as descriptors are returned as M-by-N matrix having M feature vectors and each descriptor having length N. Corresponding to each descriptor, M number of valid points is also returned. To match the features, match features function is used. To display corresponding feature points an overlay of pair of images in addition to a color-coded plot of corresponding points connected by a line, but the location is defined in the Surf point objects. [21]

4. COMPARATIVE ANALYSIS OF ALL THE THREE TECHNIQUES

Following table gives comparison of retrieval time of all types of databases with three different techniques, such as match Point, Histogram and Eigen Values.

Techniques / Database	Match Point (sec)	Histogram	Eigen values
Flower	5.357360	11.689926	2.743924
Flower	15.029603	12.321363	4.988143
Flower	6.987936	13.361342	3.311586
Flower	6.152623	12.084985	3.162717
Flower	6.943955	26.030384	5.088805
Flower	7.701922	12.703803	3.121432
Face	7.113300	11.335935	2.323375
Face	6.544435	11.324701	3.008408
Face	5.480784	17.273292	3.655586
Face	6.458972	12.654557	2.164110
Face	6.9511341	11.463735	4.656659
Face	7.245466	11.218740	4.571890
Face	6.291519	13.056973	1.921271
Face	5.202477	12.512681	2.388528
Face	6.159422	12.477321	8.896353
Vehicle	5.205715	11.218377	2.711211
Vehicle	6.210938	11.659295	3.566008
Vehicle	8.832045	11.472673	3.346753
Vehicle	7.789522	13.174998	2.197203
Vehicle	9.015264	12.895586	5.521553
Vehicle	6.083464	12.393825	3.005223
Vehicle	9.241839	16.0635481	3.049684
Vehicle	5.151585	11.022538	2.063009
Vehicle	5.835444	17.895952	2.903603
Vehicle	5.636356	10.383492	2.820357
Animal	5.851284	11.062617	3.678530
Animal	4.050715	11.664695	4.105603
Animal	6.812625	11.303356	2.770873
Animal	6.196215	10.947030	6.019186
Animal	5.685910	10.848084	2.305686
Animal	5.887391	11.004165	3.049897
Animal	7.0849334	12.512583	5.792557
Average Retrieval rate	276.5792/37 = 7.4751 sec	470.9113/37= 12.7273 sec	141.6727/37= 2.3876 sec

Table 2. Average recognition rate for the overall Face database, Animal database, vehicle database and Flower database.

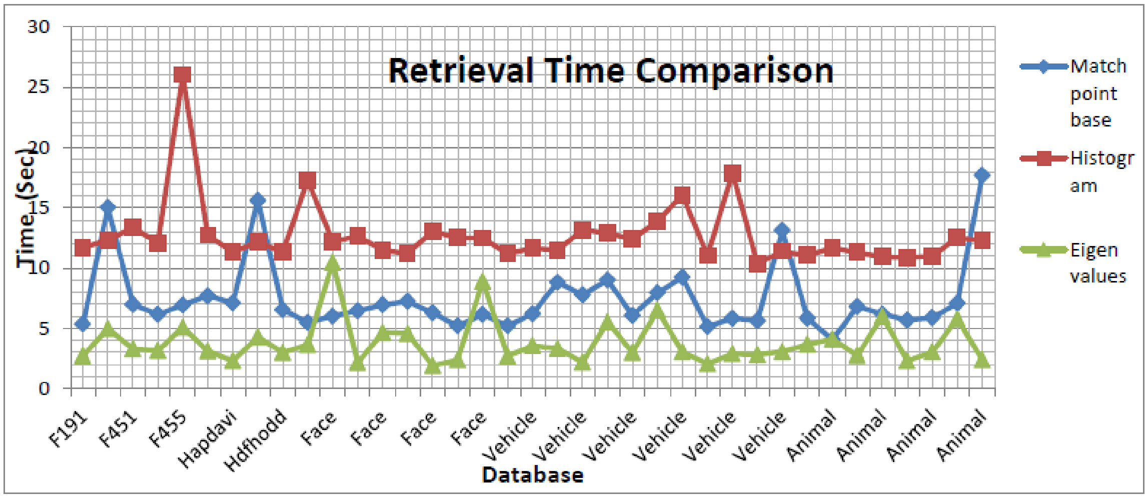


Figure 4. Graphical representation of retrieval time using three techniques.

5. CONCLUSION

This paper proposes three techniques for image retrieval from the various type of database such as human face, vehicle, animal and flower. Three techniques used here are based on match point, color histogram and eigen values. Out of these three techniques, retrieval using eigen value, found to be the best, 2.3876 seconds because only diagonal values of the images are used for comparison because of which retrieval time reduces substantially. Color Image Histogram requires maximum time, 12.7273 seconds because each pixel contributes for the construction of histogram. Lastly required retrieval time of Match Points is in between Eigen values and Histogram Techniques, 7.4751 seconds because only selected match Points contribute for the query image retrieval.

Future scope: Limitations of techniques used in this paper are number of pixels contributing the query image retrieval. In histogram technique, every pixel contributes for the plot of Histogram and hence it takes maximum time. Whereas in case of Match Points, selected pixels are used for the image retrieval and in Eigen Values only diagonal elements are used. Therefore retrieval time is least for eigen value technique, maximum for Histogram and for Match Point, retrieval time is in between Eigen Value and Histogram. In order to improve the retrieval time, Wavelets and Multi Resolution Analysis (MRA) can be used. This will result in improvement in directional information and retrieval efficiency. Also can be used identify unknown objects.

ACKNOWLEDGEMENTS

Author would like to thank to a technical paper, Relevance Feedback Techniques in Interactive Content Based Image Retrieval by Yong Rui, Thomas Huang and Sharad Mehrotra, from where my mind trigger with the idea of CBIR which is my research topic. I would like to thank my mentor, my guide Dr. G. K. Kharate, whose guidance could help me to realize the idea of CBIR.

REFERENCES

[1] Jain, "Content –Based Image Retrieval at the end of the Early Years," IEEE Transactions on Pattern analysis and Machine Intelligence, vol. 22, no. 12, pp. 1349-1380 , Dec. 2000.

[2] Belongie S., Carson C., et al., "Color - and Texture Based Image Segmentation using EM and its Application to Content –Based IImage Retrieval," in Proceedings of 8th International Conference on Computer Vision, 1998.

[3] Manesh Kokare, B .N.Chatterji and P.K.Biswas, "A Survey on current content based Image Retrieval Methods," IETE Journal of Research, vol. 48, no. 3 &4, pp. 261-271, 2002.

[4] Smith. J and Chang S. F, "Visual SEEK: a fully automated content-based image query system," in Proceedings of ACM Multimedia 96, pp. 87-98, 1996.

[5] S. Satini, R. Jain, " Similarity Measures," IEEE Transactions on pattern analysis and machine Intelligence, Vol.21, No.9, pp.871-883, Sept. 1999.

[6] X. Wan and C. C. J. Kuo, "Color distribution analysis And quantization for image retrieval," in SPIE proceedings, vol. 2670, February 1996.

[7] B. S. Manjunath and W.Y. Ma, "Texture Feature for Browsing and Retrieval of Image Data," IEEE Transactions on Pattern Analysis and Machine Intelligence, vol. 8, no. 8, 1996.

[8] Jain. A. K, Vailaya. A, "Image retrieval using color and shape," Pattern Recognition 29 (8),pp. 1233-1244,1996.

[9] Wee C. Y. and Parmesweran R. 2008, "Sharpness measure using Eigenvalues". In proceedings of IEEE 9th international conference on Signal Processing, ICSP2008, Beiging, pp 840-843.

[10] M.Turk and A. Pentland, "Face Recognition using Eigenfaces", Proceeding of IEEE conference on Computer Vision and Pattern Recognition, pp586-591, June1991.

[11] Georgia Tech Face Database, Available at ftp://ftp.ee.

[12] Gonzalez. R.C, Woods. R.E, Eddins.S.L, Digital Image Processing using MATLAB, Pearson Education,2005, p. 482.

[13] Kim H. and Kim J, "Region- based Shape Descriptor Invarient to Rotation, Scale and Translation," Signal Processing: Image Communication, Vol. 16, pp 87-93, 2000.

[14] A. Ramesh Kumar and Saravanan, "Content Based Image Retrieval using color histogram", International Journal Of Computer Science and Information Technology, Vol. 4(2), 2013,242-245.

[15] R. Chakravarti and X. Meng, "A study of color Histogram Based Image Retrieval", 6th international conference on Information Theory: New Generations, IEEE 2009.

[16] A. Vailaya, M. A. G. Figueiredo, A. K. Jain, and H. J. Zhang, "Image classification for content based indexing", IEEE transaction on Image Processing, Vol 10, No 1, Jan 2001.

[17] J. Halfner, H. Sawhney, W. Equitz, M. Flickner, and W. Niblack, "Efficient color histogram indexing for quadratic form distance functions", IEEE transaction on Patten Analysis and Machine Intelligence, 17(7):729-736, July 1995.

[18] A. W. M. Smeulders, M. Worring, S. Santini, A. Gupta, and R. jain. "CBIR at the end of early years. IEEE Transaction on Pattern Analysis and Machine Intelligence (PAMI), 22(12): 1349-1380, 2000.

[19] G. Pass and R. Zabih, "Histogram refinement for CBIR", IEEE Workshop on Application of Computer vision, pages 96 – 102, December 1996

[20] Y. Liu, D. Zhang, G. Lu, W. Y. Ma, 2007. "A survey of Content Based Image Retrieval with high level semantics", the journal of recognition society, pp. 262 –DOI:10.1016/j.patcog.2006.04.045

[21] Computer Vision System Toolbox of Matlab

AUTHORS

Mohini Sardey is currently serving as HOD and Assistant Professor in AISSMS Institute Of Information Technology, Savitribai Phule Pune University, Pune (Maharashtra State) India. She has more than 20 years of experience in teaching field. She is PhD studnt and doing her research work under the eminent guidance of Dr. G. K. Kharate. She earned her Batchelar degree in Electronics Engineering from Amravati University and M.Tech from Government College Of Engineering, Savitribai Phule Pune University, Pune. Her primary area of research is Image Processing and Machine Vision. She is life member of IETE and ISTE.

G.K. Kharate is currently Principal, Matoshri College of Engineering and Research Centre, Nashik, Savirtribai Phule Pune University, Pune (Maharashtra), India. A PhD from the University of Pune, Dr Kharate has more than 20 years of teaching experience. He is also a fellow member of the Institution of Electronics and Telecommunication Engineers (IETE) and a life member of many other professional bodies of repute like the Indian Society for Technical Education (ISTE), the Institution of Engineers (India), and the Computer Society of India. Dr. Kharate has also published a number of articles in national and international journals of repute and organized several conferences and workshops in his areas of research.

AUTHOR INDEX