NIDS is shown in Figure1. There are two phases in the framework: off-line phase and on-line phase. The system builds patterns of intrusion in the off-line phase and detects intrusion in the on-line phase.

In the off-line phase, training data sets are fed in to the pattern builder module, which can build the patterns of intrusions. The module employs the feature selection algorithm, handles imbalanced intrusions, and builds the patterns by association rule mining with user specified minsup and minconf constraints. After mining the patterns for intrusions, the module outputs the pattern as the input of the detector module.

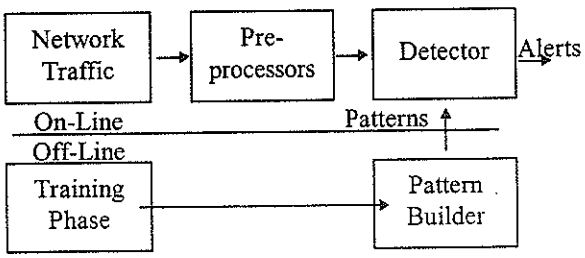

Figure 1. Framework of NIDS

In the on-line phase, the system captures the packets from network traffic. The features for each connection are constructed by the pre-processors from the captured network traffic. Then, in the detector module, the connections are classified as different intrusions or normal traffic using the patterns built in the off-line phase. Finally, the system raises an alert when it detects any intrusion.

2.2 Association Rule Mining Approach

We consider the problem of finding rules from data $D=\{t_1,t_2,..,t_n\}$, where each transaction or record $t_i \subseteq I$ and $I = \{item_1, item_2,.....,itemn\}$ is the set of items of which transactions are composed. For market-basket data items are atomic forms and for attribute-value data items have the form $\alpha_i=v_{ij}$ where a_i represents an attribute and V_{ij} a value of a_i. For attribute value data, no transaction t_i, $1\le i\le n$ may contain two items $a_i=v_{ij}$ and $a_i=v_{j,k}$, $j\neq K$. That is, each transaction may contain at most one value for each attribute. Rules take the form $x \Rightarrow y$, where $x\subset I$ and $y\in I$. Note that we limit the consequent y to a single value. While many association rule techniques allow multiple values in the consequent y, the technique we present generalize directly to multiple -value consequents and a single value with multiple elements in the consequent can be represented by multiple rules with single elements in the consequent.

In this paper, we are potentially interested in a number of properties of a rule $x \Rightarrow y$ relative to D, and the properties vary from application to application. In this, we utilize support [4], confidence [4], lift [9], and leverage [10], defined as follows:

$$Sup(x \Rightarrow y,D)= |\{i:x \subseteq t_i \cap y \in t_i\}| \text{----(1)}$$

$$Conf(x \Rightarrow y,D)= Sup(x \Rightarrow y,D)/ |\{i:x \subseteq t_i\}| \text{---(2)}$$

$$lift(x \Rightarrow y,D)=Conf(x \Rightarrow y,D)/(|\{i:y \in t_i\}|/n) \text{---(3)}$$

$$lev(x \Rightarrow y,D)= Sup(x \Rightarrow y,D)- |\{i:x \subseteq t_i\}| * |\{i:y \in t_i\}|/n \text{---- -(4)}$$

Note that the parameters $x \Rightarrow y$ and D will be omitted from these functions where they can be determined from the context.

The original association rule task [4] was to find all rules x y such that sup≥minsup and conf≥minconf, where minsup and minconf are user specified constraints.

Typically, rules will only be interesting if they represent non-trivial correlations between items. Relatively high

values of minsup and minconf usually deliver rules for which x and y are correlated when applied to the sparse data typical of market-basket analysis. However, as will be demonstrated in the experiments below, this is not the case for the dense data such KDDCup'99 data.Also, there is a serious problem that x may contain items that are independent of y, and hence potentially misleading. However, in most contexts unproductive rules will be of no interest so long as the first rule is known.

Apriori, one might expect there to be very large numbers of unproductive rules, as from every single productive rule $x \Rightarrow y$ many unproductive rules can be generated by inserting into x any arbitrary collections of unrelated items.

In order to alleviate this problem, we have used Predictive Apriori Algorithm to obtain various interestingness measures such as accuracy, lift and leverage, to discover most interesting ones out of many obtained for detecting network intrusions.

3. Experiments And Evaluation

In this section, we summarize our experimental results to build patterns for intrusion detection over the KDDCup'99 datasets. We first describe the datasets used.Finally; we evaluate our approach based on the various interestingness measures to discover most significant rules.

3.1. Dataset and Pre-Processing

Under the sponsorship of Defence Advanced Research Projects Agency (DARPA) and Air force Research Laboratory (AFRL), MIT Lincoln Laboratory has collected and distributed the datasets for the evaluation of computer network intrusion detection systems [11]. DARPA dataset is the most popular dataset used to test and evaluate a large number of IDSs. The KDD'99 dataset is a subset of DARPA dataset prepared by Stolfo and Wenke Lee [12].The data was pre-processed by extracting 41 features from the tcpdump data in the 1998 DARPA datasets. The KDD'99 dataset can be used without further time-consuming pre-processing and different IDSs can compare with each other by working on the same dataset. Therefore, we carry out our experiments on the KDDCup'99 dataset.

The KDD'99 dataset includes the full training set, the 10% training set, and the test set. The full training set has 4,898,431 connections. The 10% training set has 494,020 connections. The attacks in the data set fall into four categories [13] as shown below.

- Denial-of-Service (DoS) attacks have the goal of limiting or denying services provided to the user, computor network. A common tactic is to severely overload the targeted system (e.g. apache, Smurf, Neptune, etc.).
- Probe or Surveillance attacks have the goal of gaining knowledge of the existence or configuration of a computer system or network (e.g. Saint, PortSweep, etc.).
- U2R (User-to-Root) attacks have the goal of gaining root or super-user access on a particular computer or system on which the attacker previously had user level access. These are attempts by a non-privileged user to gain administrative privileges (e.g. Perl, Xterm, etc.).
- R2L(remote-to-Local) attack is an attack in which a user sends packets to a machine over the internet, which the user does not have access to in order to expose the machine vulnerabilities and exploit privileges which a local user would have on the

computer (e.g. Xclock, Dictionary, guest_password, etc.).

3.2. Experimental Results

For our experiments, we choose the Apriori and predictive Apriori algorithm in Weka [14], using user specified minsup and min confidence. All the experiments are carried out in Intel Pentium4 CPU, 2.8GHz, and 512MB RAM PC.

3.3. Evaluation and Discussion

We carry out the experiment over 10%KDDCup'99 attack dataset. We evaluate the performance of our system by the accuracy and interestingness measures, such as lift and conviction, apart from the user specified minsup and minconf.

Confidence (Strength): The confidence of an association rule is the proportion of the isolates that are covered by LHS of the rule that are also covered by the RHS. Values of confidence near value 1 are expected for an important association rule.

Support: The support of an association rule is the proportion of the isolates covered by LHS and RHS among the total number of isolates. Support can be considered as an indication of how often a rule occurs in a data set and as a consequence how significant a rule is.

Coverage: The coverage of an association rule is the proportion of isolates in the data that have the attribute values or items specified on the LHS of the value. Values of coverage near Value 1 is expected for an important association rule. This is otherwise treated as accuracy of the discovered association rule.

Lift $(X \rightarrow Y)$ =conf $(X \rightarrow Y)$/P(Y).an equivalent definition is: P(X, Y)/P(X) P(Y).Lift is a symmetric measure. A lift well above 1 indicates a strong correlation between X and Y.A lift around 1 says that P(X, Y) = P(X) P(Y). In terms of probability, this means that the occurrences of X and the occurrence of Y in the same transaction are independent events; hence X and Y are not correlated. Another definition can be found in [14].

Conviction $(X \rightarrow Y)$ = [1-P(Y)]/ [1-conf $(X \rightarrow Y)$]. Conviction is not a symmetric measure. A conviction around 1 says that X and Y are independent, while conviction is infinite as conf $(X \rightarrow Y)$ is tending to 1. It is to be noted that if P(Y) is high, 1-P(Y) is small. In that case, even if conf $(X \rightarrow Y)$ is strong, conviction $(X \rightarrow Y)$ may be small.

Leverage: The leverage of an association rule is the proportion of additional isolates covered by both the LHS and RHS above those expected if the LHS and RHS were independent of each other. Leverage takes values inside [-1, 1]. Values equal or under value 0, indicate a strong independence between LHS and RHS. On the other hand, values near 1 are expected for an important association rule.

From the figures 2 and 3, it can be observed that, our method is an effective and efficient one, in discovering the most significant rules and reducing the false discoveries, thus avoiding the effect of combinatorial explosion, which otherwise would have caused with many insignificant rules.

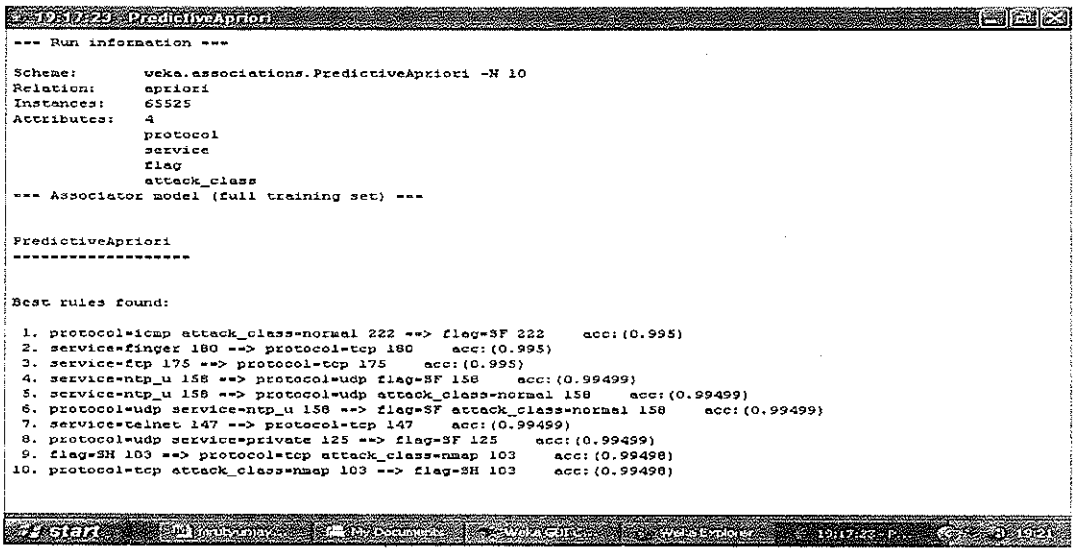

Figure 2 : Predictive Apriori to Measure the Accuracy of the Discovered Rules.

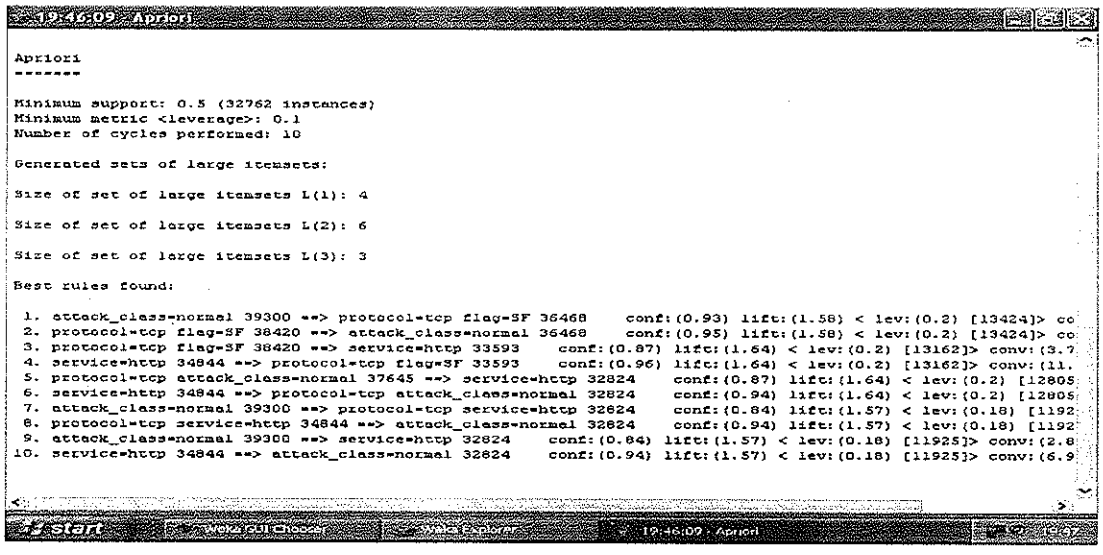

Figure 3 : Apriori with Interestingness Measures

4. Related Work

In recent years, many Data Mining -based research work have been proposed for intrusion detection.MADAMID (Mining Audit data for automated Models for Intrusion detection) [15] is one of the best known data mining projects in intrusion detection. It is an offline IDS to produce anomaly and mis-use intrusion models. Association rules and frequent episodes are applied in MADAM ID to replace hand-coded intrusion patterns and profiles with the learned rules.

ADAM (Audit Data Analysis and Mining) [16] is the second most widely known and well published project in the field. It is an on-line network-based IDS.ADAM can detect known attacks as well as unknown attacks. Association rules and classification, two data mining techniques are used in ADAM.

IDDM (Intrusion detection using data Mining techniques) [17] is a real-time NIDS for misuse and anomaly detection. It applied association rules, Meta rules, and characteristic rules. IDDM employs data mining to produce description of a network data and uses this information for deviation analysis.

In contrast to the previously proposed data mining based IDSs, we have implemented the predictive Apriori and Apriori with various interestingness measures to discover the most significant association rules.

5. Conclusion And Future Scope

In this paper, we employ Apriori algorithms in NIDSs to discover the accurate and most significant association rules. In this way, it avoids the combinatorial explosion of having so many unnecessary rules .This model enables us to find rare item rules yet without producing a huge number of meaningless rules with frequent items. The experimental results show the effectiveness in building an efficient intrusion detection model. However, Apriori algorithm with multiple minimum support criteria may be thought of as future work, so as to develop a more powerful and flexible network intrusion detection model.

References

[1] Peyman Kabiri and Ali A. Ghorbani. *"Research on Intrusion Detection and Response: A survey"*, International Journal of Network security, Vol.1, No.2, PP 84-102, sept.2005.

[2] Snort, Network Intrusion detection system,http://www.snort.org

[3] David J. hand, Heikki Manila, and Padhraic Smyth. *"Principles of data Mining"*, The MIT Press, August 2001.

[4] R.Agarwal, T. Imielinski and A. Swami. *"Mining associations between sets of items in massive databases"*, in proc. ACM-SIGMOD Int. Conf. management of Data, PP.207-216, Washington DC, May1993.

[5] G.I. Webb and S. Zhang, *"K-optimal rule discovery"*. Data mining and knowledge discovery, 10(1): PP 39-79, 2005.

[6] M.J.Zaki, *"Mining non-redundant association rules"*, Data mining and Knowledge Discovery, 9(3).PP. 223-248, 2004.

[7] H.Zhang, B.padmnabhan and A.Tuzhilin, *"On the discovery of Significant statistical quantitative rules"*, in proc. 10th. Intl. conf. KDDM (KDD-2004), PP. 374-383, ACM Press, Aug 2004.

[8] KDDCup'99 datasets, The UCI KDD Archieve,http://kdd.ics.uci.edu/databases/kddcup'99/kddcup99.html.

[9] International Business Machines, IBM Intelligent Miner User's guide, Verson1, Release1, 1996.

[10] G.Piatetsky Shapiro, *"Discovery, analysis and presentation of strong rules"*, In G. Piatetsky Shapiro and J. Frawley, Editors, KD in databases, pp.229-248, AAAI/MIT Press, Menlo Park, 1991.

[11] M.Mahoney and P. Chan. *"An analysis of the 1999 DARPA/Lincoln laboratory Evaluation data for network anomaly detection"*, proc. Of RAID, Pittsburg, USA, Sept. 2003.

[12] Charles Elkan, *"Results of the KDD'99 classifier learning"*, SIGKDD Explorations1 (2).pp.63-64, 2000.

[13] Mrutyunjaya Panda and Manas Ranjan Patra. *"Network intrusion detection using Naïve Bayes"*, International Journal of Computer Science and Network Security, Vol.7, No12, PP 258-263, Dec.2007.

[14] Weka Machine learning S/W: University of Waikato, Hamilton, NewZealand.

[15] Wenke Lee and Salvatore J. Stolfo, *"A framework for constructing features and models for NIDS"*, ACM transaction on Information and system security (TISSEC), Vol.3, No.4, Nov.2000.

[16] Davis Barbarra, Julia Couto & Sushil Jajodia.etal, *"ADAM: Detecting Intrusions by data mining"*, Proc. Of IEEE workshop on Information assurance and security, T1a31100, USA, NY, June 2001.

[17] Tamas Abraham. *"IDDM: Intrusion detection using data mining techniques"*, DSTO Electronics and surveillance research laboratory, Salisbury, Australia, May 2001.

Author's Biography

Mrutyunjaya Panda holds a Master Degree in Engineering and is presently working as an Assistant Professor in the Department of Electronics & Tele Comm. Engg., Gandhi Institute of Engg. and Technology, Gunupur, India. He has 10 years of teaching experience. Currently, he is pursuing Doctoral research in Computer Science. He has about 10 publications to his credit. His research interests include Data Mining, Network Security, Intrusion Detection and Soft Computing.

Mrutyunjaya Panda holds a Master Degree in Engineering and is presently working as an Assistant Professor in the Department of Electronics & Tele Comm. Engg., Gandhi Institute of Engg. and Technology, Gunupur, India. He has 10 years of teaching experience. Currently, he is pursuing Doctoral research in Computer Science. He has about 10 publications to his credit. His research interests include Data Mining, Network Security, Intrusion Detection and Soft Computing.

Adaptive Interpolation and Sharpening for Single Sensor Digital Camera Images

Krishnan Nallaperumal[1], S.S. Vinsley[2], C.Seldev Christopher[3]

ABSTRACT

A single sensor equipped with a Color Filter Array (CFA) is used in many Digital Still Cameras (DSC) to capture any of the three primary color components, R (red), G (green) or B (blue) on each pixel location, in order to reduce the size. This single band is used to generate the true RGB image. Demosicing is the process of interpolating the missed colors, which aims to reconstruct the missing colors as close as possible by keeping the less computational complexity. In this paper a novel adaptive edge preserving, edge directed inter-plane interpolating technique is proposed for color reproduction from blurred Bayer mosaic images. To sharpen the image, an adaptive edge directed sharpening algorithm is introduced. The proposed interpolation algorithm aims to effectively estimate the missing green component in the edge and texture regions.

Depending on the sharpness, an adaptive weighted interpolation method is introduced. Experimental results show that the proposed method performs much better than other latest techniques in terms of Color Peak Signal to Noise Ratio (CPSNR), and Structural Similarity.

[1]Professor and Head, CITE, M.S.University, Tirunelveli, E-mail: krishnan@computer.org.

[2]Narayanaguru college of Engineering, Manjalamoodu, E-mail: vinsleyss@yahoo.com.

[3]C.S.I Institute of Technology, Thovalai, E-mail: seldev@ieee.org.

Keywords : Color demosaicing, interpolation, color filter array, single sensor digital camera, sharpening.

1. INTRODUCTION

In order to reduce the hardware cost, many digital still cameras use a single sensor equipped with a color filter array to capture any of the three primary color components R, G, or B on each pixel location. Among the various suggested CFAs, the Bayer CFA pattern is the most prevalent one, where G pixels occupy half of all, R and B pixels share the others [1]. A representation of a full-color image needs all the information from the three colors on each pixel location. As a result, the missing two colors on each pixel location have to be interpolated back to get a full-color image. The process of interpolating the missing colors is called as demosaicing or color interpolation, whose main objective is to reconstruct the missing colors as close as to the original by keeping the computational complexity is very low [5] [2].

There are many interpolation methods proposed, namely nearest-neighbor replication, bilinear interpolation [14], and cubic spline interpolation, but they were not able to preserve the edge details. Other approaches are based on edge details and the reconstructions are performed in a directional way.

For reducing the various visual artifacts such as aliasing and color shifts resulted from demosaicing, camera manufacturers place a deblurring filter in the optical path [17]. The motion blur reduces both the sharpness and the

resolution of the captured image. To remove this type of blur from the image, an adaptive edge directed sharpening algorithm is introduced along with a novel adaptive weighted color interpolation algorithm. This algorithm aims to estimate the optimum weight value according to edge level for interpolation. So that, the proposed algorithm performs superbly both in textured and edge regions. In section 2, image sharpening using adaptive weighted approach is introduced, Section 3 presents the details of our adaptive weighted color interpolation technique, and Section 4 presents the comparison of simulation results in terms of both Color Peak Signal to Noise Ratio and Structural Similarity[16]. Finally, concluding remarks are made in section 5.

2. Proposed Method For Image Sharpening

To reconstruct a full-color image from CFA samples, the two missing color values at each pixel are to be estimated from neighboring CFA samples. The green plane is estimated first and the other color planes are estimated based on the interpolated value of the green plane [6]. When the green plane is processed, for each missing green component in the CFA, the algorithm performs a gradient test, to identify edge direction and then carries out an interpolation along the direction of a smaller gradient to determine the missing green component [9]. The variance of color differences can be used as a supplementary criterion to determine the interpolation direction for the green components. The parameters L^H and L^V are computed by the equations (1) and (2) to determine the horizontal or vertical gradient change in the 5x5 testing window [7].

$$L^H = \sum_{n=\pm 2}\left[\sum_{m=0,\pm 2}|R_{i+m,j+n}-R_{i+m,j}|+\sum_{m=\pm 1}|G_{i+m,j+n}-G_{i+m,j}|\right]+\sum_{n=\pm 1}\left[\sum_{m=0,\pm 2}|G_{i+m,j+n}-R_{i+m,j}|+\sum_{m=\pm 1}|B_{i+m,j+n}-G_{i+m,j}|\right] \quad (1)$$

and $L^V =$

$$\sum_{m=\pm 2}\left[\sum_{n=0,\pm 2}|R_{i+m,j+n}-R_{i,j+n}|+\sum_{n=\pm 1}|G_{i+m,j+n}-G_{i,j+n}|\right]+\sum_{m=\pm 1}\left[\sum_{n=0,\pm 2}|G_{i+m,j+n}-R_{i,j+n}|+\sum_{n=\pm 1}|B_{i+m,j+n}-G_{i,j+n}|\right] \quad (2)$$

The leading edge direction is determined by computing the ratio of the above parameters. If the ratio, $e = \max[(L^H / L^V) \text{ or } (L^V / L^H)]$ is above threshold value then it is defined as sharp edge block. The threshold value can be determined on statistical approach as in [7]. Better interpolation results are achieved for the threshold value 1.6 than others. Sharpening is performed with the knowledge of edge information as in equation (3).

$$Ssharp(i,j) = S(i,j) + \sum_m \sum_n (w_{i+m,j+n} * S_{i+m,j+n}) \quad (3)$$

Where m = - 2 :1: 2 and n = - 2 :1: 2

W be the weight matrix, for sharp edges with $L^V < L^H$, W =M1 / 10

Where

$$M1 = \begin{Bmatrix} -1 & 0 & -2 & 0 & -1 \\ 0 & 0 & 0 & 0 & 0 \\ -1 & 0 & 10 & 0 & -1 \\ 0 & 0 & 0 & 0 & 0 \\ -1 & 0 & -2 & 0 & -1 \end{Bmatrix}$$

For sharp edges with $L^V > L^H$, W =M2 / 10

Where

$$M2 = \begin{Bmatrix} -1 & 0 & -1 & 0 & -1 \\ 0 & 0 & 0 & 0 & 0 \\ -2 & 0 & 10 & 0 & -2 \\ 0 & 0 & 0 & 0 & 0 \\ -1 & 0 & -1 & 0 & -1 \end{Bmatrix}$$

For smooth regions, W =M3 / 8

Where

$$M3 = \begin{Bmatrix} -1 & 0 & -1 & 0 & -1 \\ 0 & 0 & 0 & 0 & 0 \\ -1 & 0 & 8 & 0 & -1 \\ 0 & 0 & 0 & 0 & 0 \\ -1 & 0 & -1 & 0 & -1 \end{Bmatrix}$$

3. Proposed Color Interpolation Algorithm

3.1 Weighted Edge Interpolation of Green Channel

For blocks, which are classified as sharp edge blocks, weight of sharp transition is defined [10]. And the procedure for estimating the green value is derived as in equations (4) and (5).

if $L^H < L^V$

$$g_{i,j} = Sw\left(\frac{\left(G_{i,j-1}+G_{i,j+1}\right)}{2}+\frac{\left(2R_{i,j}-R_{i,j-2}-R_{i,j+2}\right)}{4}\right)+ (1-Sw)\left(\frac{\left(G_{i-1,j}+G_{i+1,j}\right)}{2}+\frac{\left(2R_{i,j}-R_{i-2,j}-R_{i+2,j}\right)}{4}\right) \quad (4)$$

if $L^H > L^V$

$$g_{i,j} = Sw\left(\frac{\left(G_{i-1,j}+G_{i+1,j}\right)}{2}+\frac{\left(2R_{i,j}-R_{i-2,j}-R_{i+2,j}\right)}{4}\right)+ (1-Sw)\left(\frac{\left(G_{i,j-1}+G_{i,j+1}\right)}{2}+\frac{\left(2R_{i,j}-R_{i,j-2}-R_{i,j+2}\right)}{4}\right) \quad (5)$$

Sw is the weight of sharp transition, which can be determined using the ratio of color differences as in equation (6). If the value of 'e' is greater than the threshold value then weight of sharp transition is calculated using the ratio of color differences. When the value of 'e' is less than the threshold value then the weight Sm is chosen on statistical approach as 0.9, which has given the performance close to optimal.

	G_1^V	
G_1^H	**R**	G_2^H
	G_1^V	

if e < threshold & if $L^H < L^V$,

$$Sw = \frac{(R - G_1^H)(R - G_2^H)}{(R - G_1^V)(R - G_2^V)}$$

If $L^H<L^V$

$$Sw = \frac{(R - G_1^V)(R - G_2^V)}{(R - G_1^H)(R - G_2^H)} \quad (6)$$

The interpolation direction for the green components is estimated using the variance of color differences and the green component interpolation procedures defined in [7] are modified as (7), (8) and (9) to include the weight of smooth transition.

if $B\sigma_{i,j}^2 = \min\left(H\sigma_{i,j}^2, V\sigma_{i,j}^2, B\sigma_{i,j}^2\right)$ then

$$g_{i,j} = \frac{\left(G_{i-1,j}+G_{i+1,j}+G_{i,j-1}+G_{i,j+1}\right)}{4}+\frac{\left(4R_{i,j}-R_{i-2,j}-R_{i+2,j}-R_{i,j-2}-R_{i,j+2}\right)}{8} \quad (7)$$

if $H\sigma_{i,j}^2 = \min\left(H\sigma_{i,j}^2, V\sigma_{i,j}^2, B\sigma_{i,j}^2\right)$ then

$$g_{i,j} = Sm\left(\frac{\left(G_{i,j-1}+G_{i,j+1}\right)}{2}+\frac{\left(2R_{i,j}-R_{i,j-2}-R_{i,j+2}\right)}{4}\right)+ (1-Sm)\left(\frac{\left(G_{i-1,j}+G_{i+1,j}\right)}{2}+\frac{\left(2R_{i,j}-R_{i-2,j}-R_{i+2,j}\right)}{4}\right)$$

if $V\sigma_{i,j}^2 = \min\left(H\sigma_{i,j}^2, V\sigma_{i,j}^2, B\sigma_{i,j}^2\right)$ then

$$g_{i,j} = Sm\left(\frac{\left(G_{i-1,j}+G_{i+1,j}\right)}{2}+\frac{\left(2R_{i,j}-R_{i-2,j}-R_{i+2,j}\right)}{4}\right)+ (1-Sm)\left(\frac{\left(G_{i,j-1}+G_{i,j+1}\right)}{2}+\frac{\left(2R_{i,j}-R_{i,j-2}-R_{i,j+2}\right)}{4}\right)$$

3.2 Refinement on Red and Blue Interpolation

The refinement step focused on generating more consistent and close to optimal values for red and blue pixels using the values interpolated in initial step. The refinement scheme is based on green plane, and it processes the interpolated red and blue planes and to improve the quality of demosaicing result [13]. The proposed method is based on exploiting the image spectral correlation on two color planes. The weighted color difference value is calculated around a pixel under consideration of the weights along the four adjacent directions are estimated as follows:

$$\begin{aligned}
\alpha_{i-1,j} &= \left|G_{i-2,j} - G_{i,j}\right| + \left|R_{i-1,j} - R_{i+1,j}\right| \\
\alpha_{i+1,j} &= \left|G_{i+2,j} - G_{i,j}\right| + \left|R_{i+1,j} - R_{i-1,j}\right| \\
\alpha_{i,j-1} &= \left|G_{i,j-2} - G_{i,j}\right| + \left|R_{i,j-1} - R_{i,j+1}\right| \\
\alpha_{i,j+1} &= \left|G_{i,j+2} - G_{i,j}\right| + \left|R_{i,j+1} - R_{i,j-1}\right|
\end{aligned} \quad (10)$$

The color difference values at the four surrounding locations are estimated as follows:

$$\begin{aligned}
\tilde{K}_{R(i-1,j)} &= G_{i-1,j} - \tilde{R}_{i-1,j} \\
\tilde{K}_{R(i+1,j)} &= G_{i+1,j} - \tilde{R}_{i+1,j} \\
\tilde{K}_{R(i,j-1)} &= G_{i,j-1} - \tilde{R}_{i,j-1} \\
\tilde{K}_{R(i,j+1)} &= G_{i,j+1} - \tilde{R}_{i,j+1}
\end{aligned} \quad (11)$$

The weights are then assigned to the four adjacent color difference values, as follows:

$$\tilde{K}_{R(i,j)} = \frac{\frac{\tilde{K}_{R(i-1,j)}}{1+\alpha_{i-1,j}} + \frac{\tilde{K}_{R(i+1,j)}}{1+\alpha_{i+1,j}} + \frac{\tilde{K}_{R(i,j-1)}}{1+\alpha_{i,j-1}} + \frac{\tilde{K}_{R(i,j+1)}}{1+\alpha_{i,j+1}}}{\frac{1}{1+\alpha_{i-1,j}} + \frac{1}{1+\alpha_{i+1,j}} + \frac{1}{1+\alpha_{i,j-1}} + \frac{1}{1+\alpha_{i,j+1}}} \quad (12)$$

Similar computation is carried out to estimate the color difference $\tilde{K}_{B(i,j)}$ The red and blue pixel values are then refined using the weighted color difference values.

$$\begin{aligned}
\tilde{R}_{i,j} &= \tilde{G}_{i,j} - \tilde{K}_{R(i,j)} \\
\tilde{B}_{i,j} &= \tilde{G}_{i,j} - \tilde{K}_{B(i,j)}
\end{aligned} \quad (13)$$

Thus the weighted color difference value has generated more consistent and optimal for interpolation.

4. Simulation Results

To evaluate the performance of this proposed color interpolation method, simulation was carried out with twelve 24 bit digital color images used [15].

Figure 1 shows the tested images. The effectiveness of proposed sharpening and interpolation algorithm is shown in Figure 2 .

Figure 1: Set of Tested Images
(images are numbered from 1 to 12 in the order of left-to-right and top-to-bottom)

4.1 CPSNR Comparison

The CPSNR was used as a measure to quantify the performance of the interpolation methods[12]. It is calculated as in equation (14).