

A Promising technique for feature extraction and detection of faces using Sparsity - Enforcing Method

Banurekha.K.M
Department of MCA,
AVC College of Engineering,
Mayiladuthurai, India.

Nelgadevi.M
Department of CS,
Rani Anna Govt. College,
Tirunelveli, India.

Vijayaselvarani.S
Department of CSE,
Bhajarang Engineering College,
Chennai,India

Abstract: This paper proposes a sparsity enforcing based face detection with three kinds of classifiers. The feature extraction is done on the basis of rectangular feature from the integral image of the face. Sparsity enforcing method is applied for the dimension reduction on rectangular features from the generated Integral image. The performance of the classifier such as Support Vector Machines, Bayesian and AdaBoost are analyzed with the proper testing procedure. The analysis is also made, how the classifier performs according to the training sample size. Experimental results on three widely used face databases are presented to demonstrate the efficiency of the proposed approach.

Keywords: Support Vector Machines, Sparsity – Enforcing, Bayesian, AdaBoost

1. INTRODUCTION

The rapidly expanding research in face processing is based on the premise that information about a user's identity, state, and intent can be extracted from images, and that computers can then react accordingly. In the last five years, face and facial expression recognition have attracted much attention though they have been studied for more than 20 years by psychophysicists, neuroscientists, and engineers. Facial expression, occlusion, and lighting conditions also change the overall appearance of faces. The goal of facial feature detection is to detect the presence and location of features, such as eyes, nose, nostrils, eyebrow, mouth, lips, ears, etc., with the assumption that there is only one face in an image. It is worth mentioning that many papers use the term "face detection," but the methods and the experimental results only show that a single face is localized in an input image.

While numerous methods have been proposed to detect faces in a single image of intensity or color images, many were unaware of any surveys on this particular topic. [8] discuss that in general Face recognition techniques can be broadly divided into three categories based on the face data acquisition methodology: methods that operate on intensity images; those that deal with video sequences; and those that require other sensory data such as 3D information or infra-red imagery. Among the face detection methods, the ones based on learning algorithms have attracted much attention recently and have demonstrated excellent results. Since these data

driven methods rely heavily on the training sets, this paper also discuss several databases suitable for this task. The classes of face and nonface images are decidedly characterized by multimodal distribution functions and effective decision boundaries are likely to be nonlinear in the image space. S.T.Gandhe,et.al have disussed about the Principal Component Analysis, Discrete Wavelet Transform Cascaded with Principal Component Analysis, Contour Matching and Isodensity Line Maps Cascaded with Hopfield Neural Network in [11].M. Meenakshi has proposed a Digital Signal Processor (DSP)-based Prototype facial recognition and verification system in [2]. In [3] Siba Shankar Rout confirms that fixed point implementation (Integer Wavelet Transform) is best candidate than floating point implementation (Classical Wavelet Transform).A Combination of PCA-SVM (Principle component analysis-Support Vector Machine) and GA (Genetic Algorithm) is proposed by Rakesh Kumar,et.al in [4]. Artificial Neural Network is been tested for face Recognition System by S.Adebayo, et.al in [1]. In [17] A non-linear support vector machine is used to determine whether or not a face is contained within the observation window.[6][12][7] talks about Feature Selection via Sparse Approximation.[5] talks about a two-in-one portable low-cost dsPIC30F3013 digital signal processing microcontroller based system for real time face recognition and signature verification. A Survey is made on Detecting Faces in Images regardless of its three-dimensional position, orientation, and lighting conditions. Such a problem is analyzed in [14].To handle profile views

and rotated faces, a decision tree is trained to determine the viewpoint class (such as right profile or rotated 60 degrees) for a given window of the image being examined in [13][9]. Asymmetric AdaBoost and a Detector Cascade for fast face detection is used in domains where the distribution of positive and negative examples is highly skewed. This is been proposed by Paul Viola and Michael Jones in [16]. Image segmentation based on skin color, features extracted from the two dimensional discrete cosine transform (DCT), and self-organizing maps (SOM) is proposed in [10]. [15] deals with Feature based face detection using Gaussian derivative filters. To be effective, either classifiers must be able to extrapolate from a modest number of training samples or be efficient when dealing with a very large number of these high-dimensional training samples.

2. Face Extraction and Detection

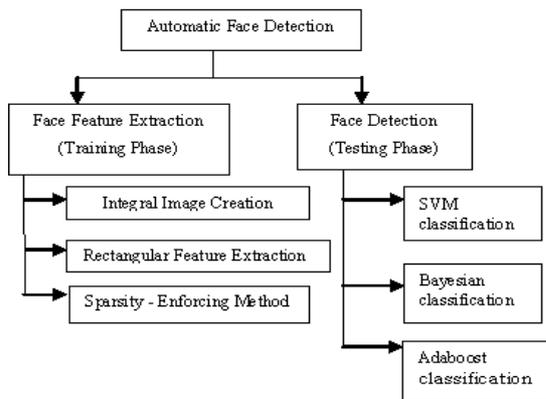


Figure. 1 Architecture Design of face extraction and detection

2.1 Integral Image

Using the integral image any rectangular sum can be computed in four array references. Clearly the difference between two rectangular sums can be computed in eight references. Since the two-rectangle features defined above involve adjacent rectangular sums they can be computed in six array references, eight in the case of the three-rectangle features, and nine for four-rectangle features.

Rectangle features can be computed very rapidly using an intermediate representation for the image which is the integral image. The integral image at location x, y contains the sum of the pixels above and to the left of x, y , inclusive:

$$ii(x, y) = \sum_{x' \leq x, y' \leq y} i(x', y')$$

where $ii(x, y)$ is the integral image and $i(x, y)$ is the original image. Using the following pair of recurrences:

$$s(x, y) = s(x, y - 1) + i(x, y) \quad (1)$$

$$ii(x, y) = ii(x - 1, y) + s(x, y) \quad (2)$$

(where $s(x, y)$ is the cumulative row sum, $s(x, -1) = 0$, and $ii(-1, y) = 0$) the integral image can be computed in one pass over the original image.

Since the two-rectangle features defined above involve adjacent rectangular sums they can be computed in six array references, eight in the case of the three-rectangle features, and nine for four-rectangle features.

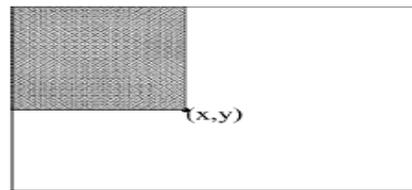


Figure 2: The value of the integral image at point (x, y) is the sum of all the pixels above and to the left.

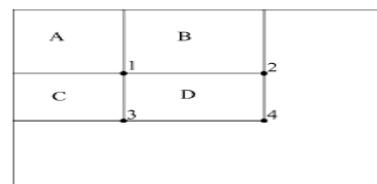


Figure 3: The sum of the pixels within rectangle D can be computed with four array references.

The value of the integral image at location 1 is the sum of the pixels in rectangle A .

The value at location 2 is $A + B$, at location 3 is $A + C$, and at location 4 is $A + B + C + D$.

The sum within D can be computed as $4 + 1 - (2 + 3)$.

In the case of convolution, if the derivative operator is applied both to the image and the kernel the result must then be double integrated:

$$f * g = \iint (f' * g')$$

Convolution can be significantly accelerated if the derivatives of f and g are sparse (or can be made so). A similar insight is that an invertible linear operation can be applied to f if its inverse is applied to g :

$$(f') * \left(\iint g \right) = f * g$$

Viewed in this framework computation of the rectangle sum can be expressed as a dot product,

i-r

where i is the image and r is the box car image (with value 1 within the rectangle of interest and 0 outside). This operation can be rewritten

$$i * r = \left(\iint i \right) \cdot r^N$$

The integral image is in fact the double integral of the image (first along rows and then along columns). The second derivative of the rectangle (first in row and then in column) yields four delta functions at the corners of the rectangle. Evaluation of the second dot product is accomplished with four array accesses.

2.2 Sparsity Feature Extraction

The number of features is much larger than the dimension of the training set, so that the system is hugely under-determined. Because of the redundancy of the feature set, the collinearities responsible for severe ill-conditioning is considered. Both difficulties call for some form of regularization and can be obviated by turning the problem. A Function is Sparse if most coefficients is zero.

The face detection procedure classifies images based on the value of simple features. There are many motivations for using features rather than the pixels directly. The most common reason is that features can act to encode ad-hoc domain knowledge that is difficult to learn using a finite quantity of training data. For this system there is also a second critical motivation for features: the feature-based system operates much faster than a pixel-based system.

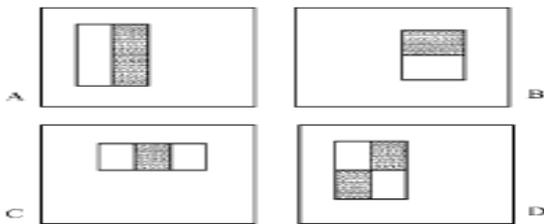


Figure 4: Example rectangle features shown relative to the enclosing detection window.

The sum of the pixels which lie within the white rectangles are subtracted from the sum of pixels in the grey rectangles. Two-rectangle features are shown in (A) and (B). Figure (C) shows a three-rectangle feature, and (D) a four-

rectangle feature. In this section, description is made on the basic algorithm on which our feature selection method is built upon. Restriction is done in the case of a linear dependence between input and output data, which means that the problem can be reformulated as the solution of the following linear system of equations:

$$g = Af \tag{1}$$

where $g = (g_1, \dots, g_n)^T$ is the $n \times 1$ vector containing output labels, $A = \{A_{ij}\}, i=1, 2, \dots, n;$

$j=1, 2, \dots, p$ is the $n \times p$ matrix containing the features j for each image i and $f = (f_1, \dots, f_p)^T$ the vector of the

unknown weights to be estimated. Since in the present context the dimensions of A are large, even if $n = p$, the usual approaches for solving the algebraic system turn out to be unfeasible. Moreover, typically the number of features p is much larger than the dimension n of the training set, so that the system is hugely under-determined. Classical regularization such as the so-called ridge regression (also referred to as Tikhonov's regularization) uses a quadratic penalty, typically the L^2 -norm of the vector f :

$$\|f\|_2 = \sum_j |f_j|^2$$

Such quadratic penalties, however, do not provide feature selection in the sense that the solution of the resulting penalized least-squares problem will typically yield a vector f with all weights f_j different from zero. This is the reason why the replacement of quadratic penalties by sparsity-enforcing penalties has been advocated in recent literature; such penalty should enforce automatically the presence of (many) zero weights in the vector f . The L^1 norm of f , which is the sum of

the absolute values of the weights, i.e., $\|f\|_1 = \sum_i |f_i|$ is a suitable penalty in order to enforce sparsity while preserving convexity of the optimization problem. Hence, instead of (1), the following penalized least-squares problem is solved.

$$f_L = \arg \min_f \{ \|g - Af\|_2^2 + 2\tau \|f\|_1 \} \tag{2}$$

where T is a positive parameter regulating the balance between the data misfit and the penalty (the so-called regularization parameter).

In variable selection problems, this parameter also allows to vary the degree of sparsity (number of true zero weights) of the vector f . After problem (2) is usually referred to as "Lasso regression" (an acronym for Least Absolute Shrinkage and Selection Operator). Whereas ridge regression

solutions depend linearly on the output vector g , the L^1 -norm penalty makes the dependence of the lasso solutions on nonlinear. Hence, the computation of L^1 -norm penalized solutions is more difficult than with L^2 -norm penalties. In the special case where $A^T A$ is the identity matrix the solution of (2) is easily seen to be $f_L = S_\tau(A^T g)$ where S_τ is the following “soft-thresholder,” widely used in wavelet-based denoising schemes, and acting component wise on a vector h

$$(S_\tau h)_j = S_\tau h_j = \begin{cases} h_j - \tau \operatorname{sgn}(h_j), & \text{if } |h_j| \geq \tau \\ 0, & \text{otherwise.} \end{cases} \quad (3)$$

Hence, the parameter appears as a threshold value, under which a component is set to zero. The number of selected features (nonzero weights) is controlled by the threshold T .

When $A^T A$ is different from the identity, there is no longer a closed-form expression for the minimizer (2) and several numerical strategies have been proposed in the literature to solve the corresponding and rather cumbersome nonlinear optimization problem. In this paper a simple iterative strategy, namely the following scheme is adopted

$$f_L^{(t+1)} = S_\tau[f_L^{(t)} + A^T(g - Af_L^{(t)})] \quad t = 0, 1, \dots \quad (4)$$

with an arbitrary initial vector $f_L^{(0)}$.

In the absence of soft-thresholding (or equivalently for $T=0$), this scheme is sometimes referred to as the Landweber iteration and converges to the generalized solution (minimum-norm least-squares solution) of (1).

Algorithm Specialization: It starts by considering the problem in the form (1) where A is the n by p matrix of the processed image data, the entry A_{ij} representing the j^{th} rectangle feature obtained from the image labeled by i . The data matrix is manipulated in order to center the features values around their median and then normalized by dividing its entries by a number slightly larger than its largest eigen value (this is done in order to guarantee convergence of the iterative scheme).

Since working in a binary classification setting this paper associate to each image a label $g_i \in \{-1, 1\}$. Each entry of the unknown vector f is associated to one feature.

Performing feature selection is equivalent to looking for a

sparse solution $f = (f_1, \dots, f_p)^T$ of (1), i.e., for a

weight vector that has many zero entries. This is done through the use of the iterative algorithm (4). Features corresponding to nonzero weights f_i are retained as relevant to discriminate between the two classes. After checking empirically that the choice of the initialization vector was not a crucial issue, it chose to always initialize the weight vector f with zeros:

$$f^{(0)} = 0^T$$

Once the matrix is correctly prepared, the iterative scheme is very simple to implement, which provides a pseudo-code describing the matrix construction, centering and normalization, and the iterative scheme (4). The following stopping rule are used for the iteration according to the stability of the solution reached: at the t th iteration it evaluate

$$\|f^{(t)} - f^{(t-1)}\|_2$$

if this quantity is smaller than some prescribed threshold T (which it choose to be a fraction of the value of the norm

$$\|f^{(t)}\|_2, T = \|f^{(t)}\|_2/100)$$

during 100 consecutive iterations, then the obtained solution is stable and concluded to stop the iterative process.

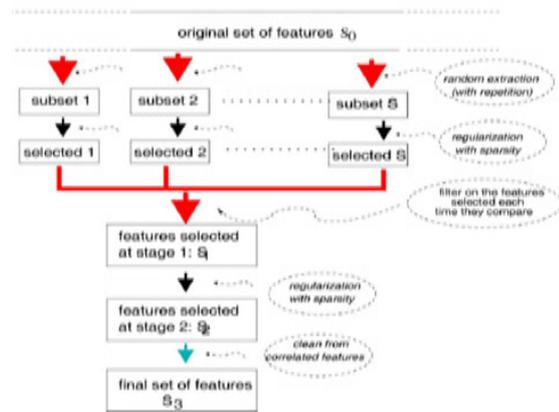


Figure 5: Flow of feature sets from the original description to the final set (dashed balloons indicate the different filtering operations performed).

A different approach was investigated and implemented, which is based on a resampling of the features set aiming at splitting the original problem into many problems of smaller size. It consists of building feature subsets by extracting each time m features from the original

set of size $p(m \ll p)$, thus obtaining smaller linear subproblems of the type

$$A_s f_s = g \quad s = 1, \dots, S$$

where A_s is a submatrix of A containing the columns relative to the features in s ; f_s is defined accordingly. The number of subproblems and their size are needed to choose. Each time extraction is done with replacement a subset of the original features. The subset size should be big enough to be descriptive, but small enough to produce an easy-to-handle matrix.

The Feature Selection Procedure

- Step 1: Find the face features for each image in the gallery.
- Step 2: Label each image (feature set) with proper class label.
- Step 3: For each image feature set, find the median value which is normalized by the iterative approach.
- Step 4: Find highest eigen values from the feature set as λ_{\max} .
- Step 5: Normalize the feature set as well as class label with λ_{\max} .
- Step 6: Apply threshold landweber for the finalized feature to represent face or non face using iterative threshold mode with the help of $f = f + A'(g - a * f)$.
- Step 6.1: Check if feature \geq threshold if so assign

$$\text{feature} = \text{feature} - \text{threshold} * \text{sign}(\text{feature})$$

otherwise set the the feature as null.

In **Feature Storing** process, the trained features are stored.

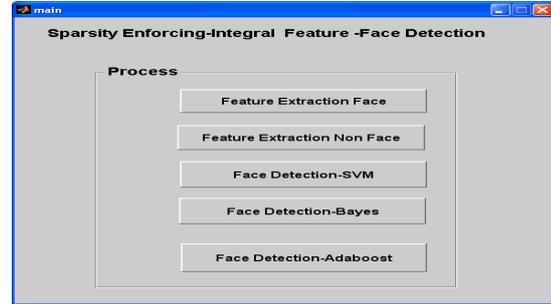


Figure 6: Overall process of feature extraction and face detection

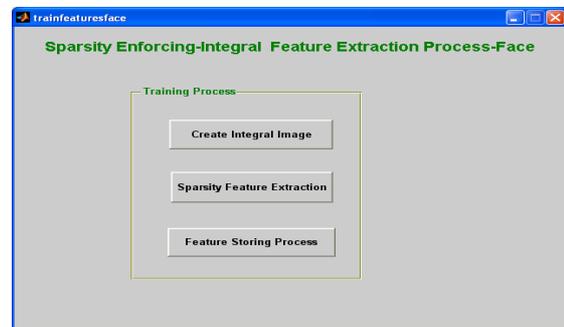


Figure 7: Feature extraction process on faces

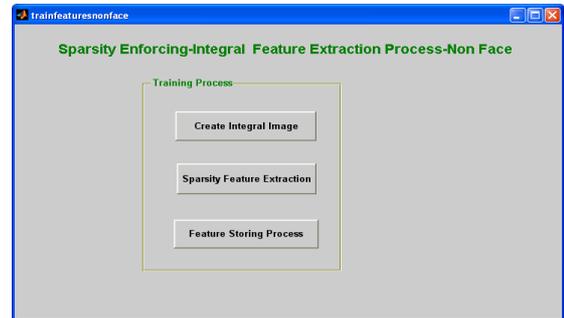


Figure 8: Feature extraction process on non-faces

2.3 Face Detection Using SVM Classifier

It is a binary classification method that finds the optimal linear decision surface based on the concept of structural risk minimization. The decision surface is a weighted combination of elements of the training set. These elements are called support vectors and characterize the boundary between the two classes. Given a set of N examples:

$$(x_1, y_1), \dots, (x_i, y_i), \dots, (x_N, y_N) \quad x_i \in \mathbb{R}^N, y_i \in \{-1, 1\}$$

In case of linear separable data, maximum margin classification aims to separate two classes with hyperplane that maximizes distance of supports vectors. This hyperplane is called OSH (Optimal Separating Hyperplane). OSH can be expressed as in equation:

$$f(x) = \sum_{i=1}^N \alpha_i y_i (x_i^T x) + b$$



Figure 9: Face detection using SVM classifier

2.4 Face Detection Using Bayesian Classifier

Consider each data instance to be an n-dimensional vector of attribute values:

$$X = (x_1, x_2, x_3, \dots, x_n) \quad (1)$$

In a Bayesian classifier which assigns each data instance to one of m classes C_1, C_2, \dots, C_m , a data instance X is assigned to the class for which it has the highest posterior probability conditioned on X , i.e. the class which is most probable given the prior probabilities of the classes and the data X . That is to say, X is assigned to class C_i if and only if $P(C_i|X) > P(C_j|X)$ (2)

For the data, $m = 2$, since there are two classes. Bayes Theorem says,

$$P(C_i|X) = P(X|C_i)P(C_i)P(X) \quad (3)$$

Since $P(X)$ is a normalizing factor which is equal for all classes, only maximise the numerator $P(X|C_i)P(C_i)$ in order to do the classification. Both the values can be estimated. $P(X|C_i)$ and $P(C_i)$ are needed from the data, for building the classifier.

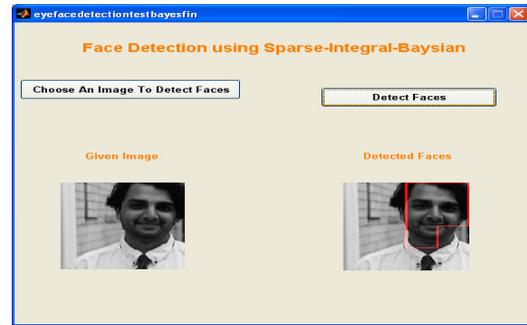


Figure 10: Face detection using Bayesian Classifier

2.5 Face Detection Using Adaboost (AB) Classifier

AB is a linear classifier with all its desirable properties. Its output converges to the logarithm of likelihood ratio. It has good generalization properties. It is a feature selector with a principled strategy (minimisation of upper bound on empirical error).

AB close to sequential decision making (it produces a sequence of gradually more complex classifiers).

AdaBoost is an algorithm for constructing a "strong" classifier as linear combination

$$f(x) = \sum_{t=1}^T \alpha_t h_t(x)$$

of "simple" "weak" classifiers

2.5.1 Terminology

$h_t(x)$ is "weak" or basis classifier, hypothesis, "feature"

$H(x) = \text{sign}(f(x))$ is "strong" or final classifier/hypothesis

The $h_t(x)$ can be thought of as features. Often (typically) the set $H = \{h(x)\}$ is infinite.



Figure 11: Face detection using Adaboost Classifier

3. RESULT ANALYSIS

The facial images from the database MIT-CMU dataset are used. From that, a training set for faces as well as for non faces of size 32 x 32 was developed. The sample faces and non faces for feature extraction in training section are shown below

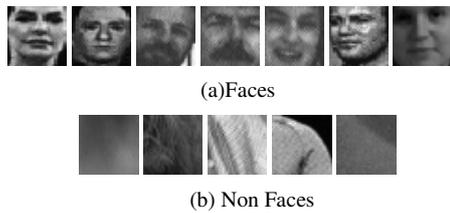
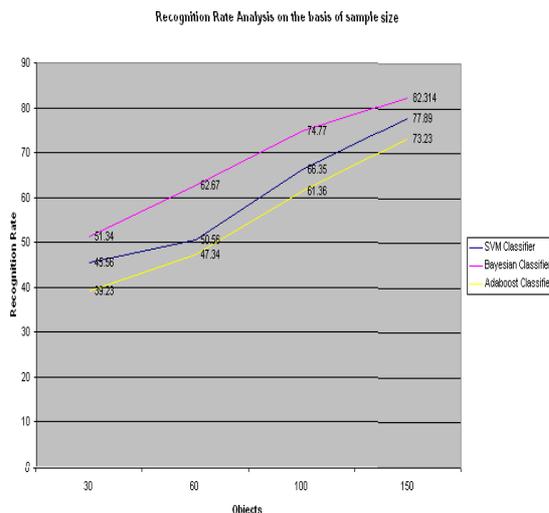


Figure 12: Sample faces and non faces for training section

3.1 Analysis on Detection Rate

From the below result, the approach of sparsity enforcing rectangular feature extraction on the basis of Bayesian classifier performs well than up to 4.424%. Bayesian classifier performs well on low and high resolution images. The performance also depends on the size of feature vector for training the detection rate which increases along the sample size and the error rate analysis is also properly maintained according to the sample size.



The performance of the Face detection using Sparsity-Enforcing Method for Learning Face Features is estimated with three modes of classifiers.

Table 1: Analysis on Detection Rate using Adaboost, SVM and Bayesian Classifier

Approaches	Classification Rate	Error Rate
Sparsity-Enforcing Approach on Rectangular Feature and Adaboost Classifier	74.21%	0.39
Sparsity-Enforcing Approach on Rectangular Feature and SVM Classifier	77.89%	0.33
Sparsity-Enforcing Approach on Rectangular Feature and Bayesian Classifier	82.314%	0.20

Table 2. Performance analysis on Face Detection using Adaboost, SVM and Bayesian Classifier with different Sample Size

Approaches	30	60	100	150
Sparsity-Enforcing Approach on Rectangular Feature and Adaboost Classifier	39.23	47.34	61.36	73.23
Sparsity-Enforcing Approach on Rectangular Feature and SVM Classifier	45.56	50.56	66.35	77.89
Sparsity-Enforcing Approach on Rectangular Feature and Bayesian Classifier	51.34	62.67	74.77	82.31

4. CONCLUSION

In this paper, the process of face detection using sparsity enforcing approach on face features with three kinds of analyzer are used to evaluate its performance on classification accuracy. The process of detecting face is done through three stages. In the first stage, for computational reasons, to split a big problem into a number of smaller problems involving randomized blocks of features was proposed in order to select a smaller but still highly redundant set of features; this stage could be further optimized by exploiting parallel computation. The second stage is based on applying again the selection algorithm to the features resulting from the first stage. The third stage is based on finding a very small set of uncorrelated features that is suitable for real-time processing to the price of a very limited decrease in performance. The latter description on a real-time face detection system was tested based on a cascade of SVM, Bayesian and AdaBoost classifiers. Texture features based detection and detection on the skin color based approach for the color image analysis can also be an enhancement to this work.

5. REFERENCES

- [1] S.Adebayo,Daramola,O.Sandra Odeghe,“Efficient Face Recognition System using Artificial Neural Network” , International Journal of Computer Applications Volume 41, No.21, 2012.
- [2] M. Meenakshi,“Real-Time Facial Recognition System Design, Implementation and Validation”, Journal of Signal Processing Theory and Applications, vol.1, pg: 1-18,2012.
- [3] Siba Shankar Rout,“Integer Wavelet-based PCA for Face Recognition”,International Journal of Computer Applications ,Volume 54, No.7, 2012.
- [4] Rakesh Kumar Yadav,A. K. Sachan,D. Rai,“Research and Applications of Optimal Face Recognition System”, International Journal of Computer Applications ,Volume 44, No.3, 2012.
- [5] Ibiyemi,T.S, Ogunsakin,J, Daramola S.,“A Bi-Modal Biometric Authentication by Face Recognition and Signature Verification”,International Journal of Computer Applications, Volume 42,No.20,2012.
- [6] Yixiong Liang, Lei Wang,Yao Xiang, and Beiji Zou ,“Feature Selection via Sparse Approximation for Face Recognition”,IEEE TRANS. PATTERN ANAL. MACH. INTELL., JULY 2010.
- [7] Augusto Destrero,Christine De Mol,Francesca Odone, and Alessandro Verri,“A Sparsity- Enforcing Method For Learning Face Features”2009.
- [8] Rabia Jafri and Hamid R. Arabnia,“A Survey of Face Recognition Techniques”,Journal of Information Processing Systems,Vol.5, No.2, June 2009.
- [9] Margarita Osadchy,Yann Le Cun,Matthew L. Miller ,“Synergistic Face Detection and Pose Estimation with Energy-Based Models”2007.
- [10] Abdallah S. Abdallah, A. Lynn Abbott, and Mohamad Abou El-Nasr ,“A New Face Detection Technique using 2D DCT and Self Organizing Feature Map”2007.
- [11] S.T.Gandhe, K.T.Talele, A.G.Keskar,“Intelligent Face Recognition Techniques: A Comparative Study” International Journal on Graphics,Visio and Image Processing ,Volume 7, Issue 2, August, 2007.
- [12] Allen Y. Yang, John Wright,Yi Ma, and S. Shankar Sastry,“ Feature Selection in Face Recognition: A Sparse Representation Perspective”IEEE TRANS. PAMI, AUGUST 2007.
- [13] Jones M, Viola P , “Fast Multi-view Face Detection”, 2003.
- [14] Ming-Hsuan Yang, David J. Kriegman, and Narendra Ahuja, “Detecting Faces in Images: A Survey” IEEE transactions on pattern analysis and machine intelligence, VOL. 24, NO. 1,2002.
- [15] Feature based face detection using Gaussian derivative filters, 2002.
- [16] Paul Viola and Michael Jones, “Fast and Robust Classification using Asymmetric AdaBoost and a Detector Cascade”, 2002.
- [17] Sami Romdhani, Philip Torr,Bernhard Scholkopf, “Andrew Blake Computationally Efficient Face Detection”, 2001.

QR Code Library on the Base of Software Reuse Approach

Amandeep Kaur
Department Of Computer Science
Lala Lajpat Rai College Of Engineering &
Technology,Moga
India

Harwinder Sohal
Department Of Computer Science
Lala Lajpat Rai College Of Engineering &
Technology, Moga
India

Abstract: Software Reuse is an approach of reuse the previously old concepts or objects into a new environment or situation and represent something new one. This representation easily use for future reference. Software Reuse model was planned, analyzed, and categorized before creating a software so that any changes persist or need to embed any extra feature then that should be introduced easily and with less complexity by using prebuild assets i.e software system is developed such that it can be reused again. The Focus of this paper to analyze how to create QR code library by using new technology like android and by using software reuse approach. How software reuse approach work in QR code library. Certain approaches such as design pattern, Aspect Oriented Integration, Generator Reuse, Object Oriented Programming Structure and Software Reuse Libraries, framework integration etc are keeping in mind while developing software reuse System. This approach increasing productivity, saving time and reducing cost of software development and minimize schedule overruns.

Keywords: QR code, Software Reuse, Android Framework, Barcode.

1. INTRODUCTION

Software is the use of existing assets in some form within the software development process. Software assets are products and by-products of software development life cycle and include software components, test suites, design and documentation. Software Reuse Concept was first introduced in the 1968 at N.A.T.O conference by Doggle McIlroy. Basically this conference was introduced to have focus on Software Crisis; it is referred to as the problem of development of big and reliable software in a cost effective way. A seminal report was introduced in the conference: Mass Produced Components by Douglas McIlroy. Douglas McIlroy of Bell Laboratories proposed a software reuse library which can be used again and again for developing new software [1].

Reuse is most effective when it is practiced systematically. There are various types of software reuse like systematic reuse, Compositional reuse, generative reuse. Systematic reuse is a type of reuse when reuse of assets is planned with well defined processes and life cycles. Reuse can be achieved through different modes. Compositional reuse involves constructing new software products by assembling existing reusable assets, while generative reuse involve the use of application generator to build new applications from high level descriptions. Leveraging is another term related to software reuse that involve the modification of previously developed software for a new product. Leveraging can be advantageous over creating software from scratch in that it require less time and effort. Typically software reuse involve the reuse of portion of code (e.g library subroutine) by other programmers in the same organization. A reusable process can be any information in physical or electronic form which developer may need in the process of creating software. such resources can be reused in new situation. some classes more reusable for creating software library. Reuse occur when a

developer uses a resource developed by another software developer. software reuse may be ad hoc or opportunistic [2].

2. SOFTWARE REUSE APPROACHES

The gap between the rising demands of complex software systems and ability to deliver quality software in a timely and cost effective manner keeps increasing. This has resulted in a great pressure to improve productivity and efficiency of software development. Software reuse approaches are the best way to achieve promised potential of software reuse.

Many approaches has been taken into mind while creating software reuse system such as generator reuse, Aspect orientation approach, Cots Integration, Framework Integration, program Libraries, Design Patterns, application product lines, Service Oriented. The Design Pattern represents the generic abstractions that occur across applications and show abstract and concrete objects and interactions. There are various abstract and concrete classes that can be adapted to create application system.

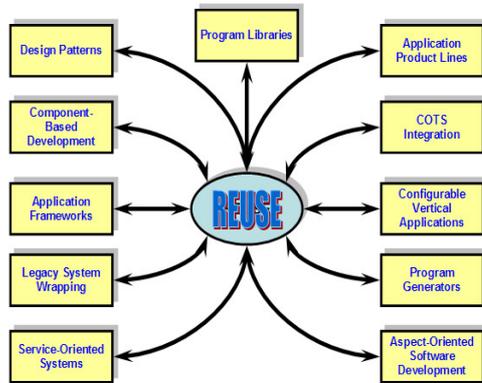


Fig 1- This diagram basically show the various approaches that could be used for creation of software and software components on the base of software reuse

3. QR CODE LIBRARY

A QR(“quick response”) code is a two dimensional bar code invented by the Japanese corporation Denso Wave.Information is decoded in both vertical and horizontal direction.In this paper we explore the structure and creation process of QR code.QR code is advancement of bar code technology.Any item you purchase these days has bar code on it.you have definitely seen these black stripes on items offered in retail store.The future of barcode is bright and exciting.The 2D barcode different from traditional barcode in its structure and properties because 2D barcode can hold more data than 1D barcode.So the manufacturer can put thousands of characters in a single 2D bar code.2D bar code using an image sensor to capture the image of barcode.QR code store huge amount of data that can be viewed anywhere,any time.QR code play an important role in education.QR code connect the user to the information quickly and easily.[3].

3.1 Structure Of QR Code

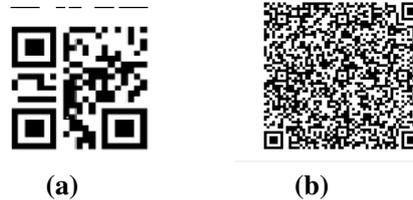
They have a matrix format.They can hold up-to 7,089 numeric characters and 4,296 alphanumeric character.They appear as a square grouping of black squares on a white background with three large square on the corner of the code.QR codes,barcodes are all systems for conveying large amount of data in a small format.QR code can contain a URL,Contact info,email address,SMS text message and even geolocation information.QR code are useful for promoting your work because they provide an easy way for public to learn more about your work.



Fig 2-Image of QR code

3.2 How QR Image Created

There are different fonts used for creating QR images.These fonts are used for creating data in the form of QR images.Before the research on fonts of QR code.I studied how barcode is created by using different fonts.QR code is extended version of Barcode.My main motive of research is how much barcode fonts reused in creating QR code code.QR code generator do the work of encoding the information.To read this information user need QR reader application to take a snapshot of the code with their device camera.This application return the decode text or web url.The QR code image Shape vary according to type or amount of data.if we encoding upto 15 alphanumeric character it does not require many pixels it look like image(a),while encoding upto 395 characters require more densely packed matrix it look like image (b).



4. PROBLEM FORMULATION

Software Reuse can lead to unbounded improvements in both development productivity and software quality.Reuse is about building software from components.The benefits will be more in case the component reused are bigger and more generalized.In this we formulate how much percentage software reuse concept used for creating QR code Library.Problem definition include various reasons that why we have chosen software reuse concept.I have research on the Library of QR code.No doubt QR code is an extended version of barcode.The fonts that are used for creating Bar code have reused for creating QR code.And I have creating QR code on android based operating system.We also discuss the similarities between QR code and Barcode.In this paper we conclude how much software reuse concept used in creating QR code.I studied how software Reuse concept increase the software productivity and how reuse reduce the software development time.And how it improve software system interoperability.It also reduce software development and maintenance cost.Reuse produces more standardized software.it provide a more powerful competitive advantage.

5. POPULARITY OF QR CODE

Mostly consumers today don't know what a QR code is- according to recent research-79% of consumers don't know what a QR code is,However 81% can identify a QR code.In 2012 most people would not know QR code from a barcode.Most are not even QR prious.QR code have been spotted on everything from building to business cards , wine bottles to tide bottles and even as tattoos.In this we first research on which devices are most popular for scanning QR code

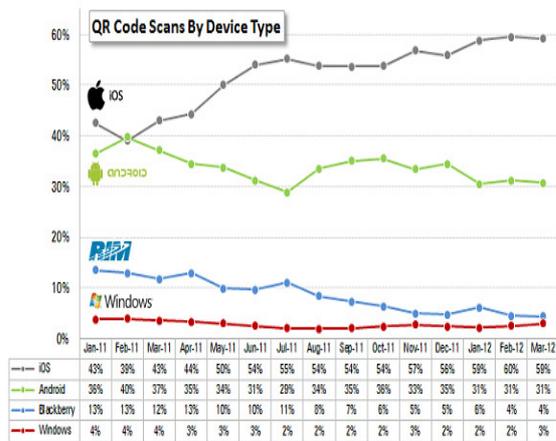


Fig 3-This Diagram show Popularity Of QR code

This Chart show iph are still most popular devices for scanning QR code.while android devices shown some narrow gap as compared to iph.The next graph show how many peoples are aware about QR code.There are many number of peoples aware about QR code.

The third graph how many countries use QR code.The foreign countries mostly use QR code.19% of American consumer have used a QR code.Of the 4 countries analyzed by the study,the US ranked first in QR code use,ahead of the UK(15%),Germany(14%) and France 12%,while on average 15 % of consumers across those countries report having used a QR code,the percentage rises to 27% among 18-34 years old.QR used as a marketing trend.

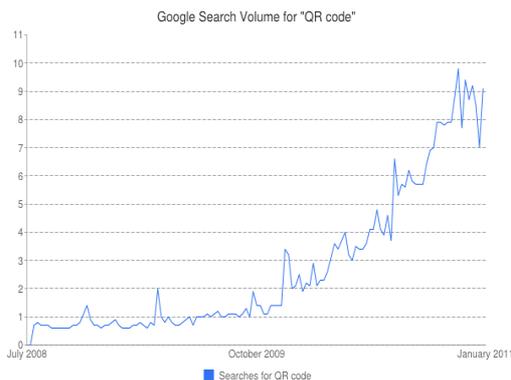


Fig 4-This Graph show how many peoples aware about QR code

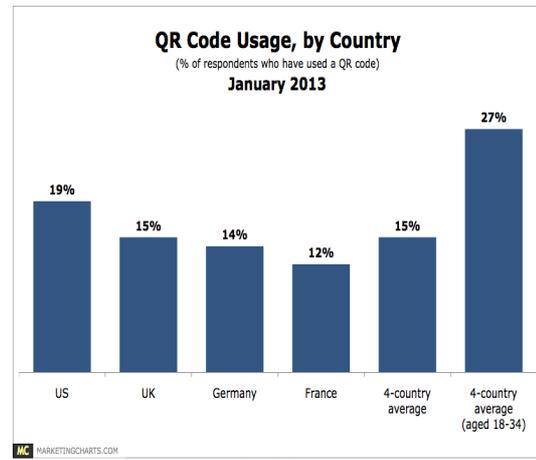


Fig 5-This Graph show how many countries use QR code

6. BARCODE Vs QR CODE

Barcode are seen on almost every product we purchase from grocery items to electronics and household items.Bar codes are one dimensional numeric codes that store upto 20 numeric characters.This allow merchants and suppliers to keep track of inventory both coming into stores and being sold.QR code are two dimensional codes storing data both horizontally or vertically.This allow QR code to hold upto 7,089 characters of data.

QR code Winner In term of data storage:-The data encoded can include numbers,alphanumeric characters symbols,text symbols such as kanji(Japanese language symbols) as well as control codes,because these codes are stored both horizontally and vertically.Infact QR codes can hold text messages,website address,contact information,phone numbers and more.In the battle of QR code and barcode,QR code are the winner in term of data storage and increased functionality.

Encoding mode	Maximum capacity
Numeric	7089 digits
Alphanumeric	4296 characters
Binary	2953
Kanji	1817

Table-1 Data Storage Capacity

Data Restoration:-Occasionally QR codes and bar codes become damaged or they may get dirty.Barcode reader will not be able to scan a damaged or dirty code.QR code can be scanned upto 30% of code words in a QR code can be restored depending upon amount of damage.Finally QR code superior in recovering lost or damage data.

Expert in Scan Position and Speed:-Barcode must be scanned in the correct position.At your local grocery store,the shopkeeper takes the items you wish to purchase and positions the barcode over the scanner.if the shopkeeper does not position the barcode correctly,the item will not scan.But QR code can be scanned from any position.This is due to the three position detection patterns located in three corner of the

code. The reader will locate these three detection patterns and know how to correctly read the code. This feature speeds up the time needed to scan objects.

Structure Appended Feature: -The data on a barcode cannot be divided up. A larger QR code can be divided into as many as 16 smaller squares. This feature allows larger QR code to be stretched out on an object. Thus, larger code printed onto a narrow area. QR code located on any object. QR code have flexibility. QR code more reliable.

More Versions and Size Of QR code As compared to Bar Code: -QR standard specifies 40 different sizes of the QR code and maximum data capacity will also vary according to size. Bar code have 24 versions including 128 fonts. The UPC (universal product code) was the first barcode symbology widely adopted. UPC as the standard barcode for product marking.

QR different in shape and size than Barcode: Barcode are created by translating the supported characters that should be displayed into combination of narrow and wide bars which are combined into a barcode. To identify the start and end of a barcode special "guard" patterns are used to indicate to the scanner and also identify what type of symbology used.

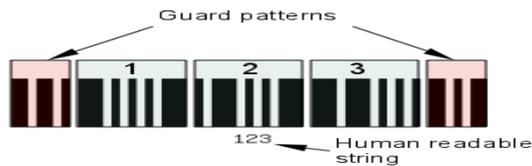


Fig-7

But the QR scanner scan the data from any direction. QR scanner decode the content within the QR code due to three specific squares that are placed in the corner of the symbol.

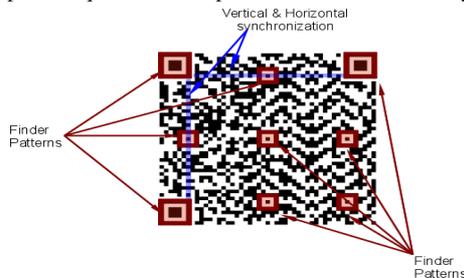


Fig-6

Finally we can say the bar code fully replaced by QR code in future. QR code are powerful because they are software. A software approach is portable, work on any device and work with any data.

7. DISADVANTAGES OF QR CODE:

Lack of Awareness: -Not everyone is aware of QR codes. And not everyone take the picture of a matrix because not everyone own a camera phone and cell phones do not include a QR reader, the software must be downloaded and tested. QR code help the user to directly connect the user with the web site that does not properly display on cell phone. Since the implementation of QR code is relatively new concept.

Expensive Smartphone and apps required: User needs to have a smartphone in order to use one. Along with the smartphone they also need a QR code reader application. Not everyone in the world own a smart phone so QR code not available to everyone.

QR code not default provided: -QR code reader are not preinstalled on most phones. It is installed by user.

8. EXPERIMENTAL SETUP:

In experimental setup, on the base of software reuse approach I have created a software reuse library for android operating system at application level which is written in java. During software development, the software programmer focuses on those reusable resources that easily adopt the new environment. My main motive in this paper to explain how we make new libraries by reuse pre-build libraries. Mobile learning is a major field of research in education. QR code is a very latest technology in mobile phones. QR code very famous day by day. The idea of creating QR code comes from bar code. There is some limitation of bar code because it does not hold much more data as compared to QR code. I do work on QR code. I have read many research papers on QR code. The implementation of QR code is relatively a new concept. QR code is extensively used in some Asian countries and is finding more and more usage to transfer medium sized information onto mobile phones where the QR codes are interrelated by first taking a photo of the barcode with the mobile and then running a QR decoding program on the cell phone.

9. CONCLUSION

The overall result and the conclusion for the software reuse is that, in today era software reuse plays a very important role for the developers while creating and developing any software or framework. Because whenever any software is developed, it is developed according to the future reference in mind, there may be some advancement or new features that may needs to add in software future, new versions for the software continue to be coming in the market or industry. Now for instance if the developers or organization does not keep in mind the software reuse they had create software with respect to the new features again and again which will result in the wastage of time, wastage of money and also wastage of resources. On the other hand if software reuse concept is keep in mind while developing software then it will save money, save time and resources because the software is designed such that it can meet the future requirements easily and properly with less time and money complexity. Simple examples are mobile platform versions, day to day new language versions, day to day new framework versions etc. Today people uses various software, with respect to time their advance versions are also in the market whether that software belongs to social networking, entertainment related, business related, sports related etc. Now what the developers do with respect to those new versions, do they use to develop it again? No they designed it such that they can use the components of the software again and make it customizable.

10. REFERENCES

- [1] Charles W. Krueger "Software Reuse" <http://www.cin.ufpe.br/~esa2/papers/abstracts/art12.htm>
- [2] Denso (2010b). About 2D Code. Retrieved June 21, 2010, from <http://www.densowave.com/qrcode/aboutqr-e.html>

- [3] Peter Kieseberg, Manuel Leithner, Martin Mulazzani, Lindsay Munroe, Sebastian Schrittwieser, Mayank Sinha, Edgar Weippl, "Qr Code Security" 2010
- [4] Thony, Retrieving Data "http://www.thonky.com/qrcodetutorial/introduction/#general-overview-of-creating-a-qr-code"
- [5] So, S. (2008). A Study on the Acceptance of MobilePhones for Teaching and Learning with a group of Preservice teachers in Hong Kong. Journal of Educational Technology
- [6] Johanes Sametinger "Software Engineering with Reuseable Components" Book publish on March 2007
- [7] ID Automation retrieving barcode library "http://www.idautomation.com/java/font-encoder.html"
- [8] Thonky QR code retrieving data http://www.thonky.com/qrcode-tutorial/introduction/
- [9] JISC (2010). JIS X 0510:2004. Retrieved May 15, 2010, from <http://www.jisc.go.jp/app/pager?id=12766> (written in Japanese)
- [10] http://mpa.ub.uni-muenchen.de/41141/1/MPRA_paper_41141.pdf (Feb 2012).

Performance Enhancement of Network Transmission by Change of Delay Parameter to Increase Throughput

Manish Giri
MIT Academy of
Engineering,
Pune, India

Manik Hendre
MIT Academy of
Engineering,
Pune, India

Abhijeet Kolte
MIT Academy Of
Engineering,
Pune, India

Kaustubh Jawalekar
MIT Academy Of
Engineering,
Pune, India

Aakash Chaudhari
MIT Academy Of
Engineering,
Pune, India

Abstract: Congestion is a major problem which really degrades network performance. Its effect on the delay is quite high which decrease the throughput in considerable amount. In this paper we have proposed a new congestion scheme which will enhance performance of network transmission by considering delay and throughput as main components. For this we are using Adaptive Explicit Congestion Notification method which allows us to control the network transmission from source to a receiver in a hierarchical manner by alterations in the receiving sending capacity. To achieve predictable average delays with adaptive explicit congestion notification would require constant tuning of the parameters to adjust to current traffic conditions. The sending and receiving ends are tuned according to its capacity and current network traffic condition once ECN (Explicit congestion notification) notification packet comes into action. The aim of this paper is to solve the parameter tuning problem of the AECN by dynamically setting up the network parameters to overcome delay in network transmission and hence to increase throughput .We will be comparing the performance of the ECN enabled system with the AECN enabled system. For this we are going to use JAVA , JNETPCAP and WINPCAP API's by which we can create TCP packets, alter the fields of headers, send and receive packets etc.

Keywords: congestion window, Explicit, Packet, Marking, Notification

1. INTRODUCTION

The accumulation of the packets in the queue results in the saturation at the end points in the communication links. The TCP/IP protocol as a primitive measure drops some packets to overcome the overflow as in the active queue management. Now if the end points are made ECN capable, the receiver sends a notification

Packet of its status .The sender then has to tune up considering the severity of the receiver ends to what extent the rate can be decreased. Also making use of ECN we can increase transmission rate if senders sends data with a capacity less than of its receiver.

1.1 Explicit Congestion Notification

ECN is an extension to the Internet Protocol and to the Transmission Control Protocol and is defined in RFC 3168 (2001). ECN allows end-to-end notification of network congestion without dropping packets. ECN is an optional feature that is only used when both endpoints support it and are willing to use it. It is only effective when supported by the underlying network on which the transmission is active. When ECN is successfully negotiated, an ECN-aware router may set a mark in the IP header instead of dropping a packet in order to signal impending congestion. The receiver of the packet echoes the congestion indication to the sender, which reduces its transmission rate as though it detected a dropped packet.

1.2. Multilevel Approach of ECN

Marking of Bits:-

AECN uses two bits that is being specified for the use of ECN, in the IP header bit 6 and 7 in the TOS octet in Ipv4, or the Traffic class octet in Ipv6 to indicate four different levels of congestion, instead of the binary feedback provided by ECN.

Table 1. ECT and CE marking

CE(Congestion Experienced) Bit	ECT(ECN Capable Transport)	Congestion state
0	0	ECN
0	1	No
1	0	Incipient
1	1	Moderate
Packet Drop		Severe

00 is used for identifying non-ECN capable packets and other combinations are used for indicating different levels of congestion which are then used to take proper action at TCP source depending on level of congestion as given in Table 1.

Sender and receiver side:-

Bit marking in IP header is reflected by receiver, to the TCP ACK. We use 3 combinations of 2 bits 8, 9 in TCP header and other combination used by source has to indicate that congestion window reduced.

Table 2. Receiver side CWR and ECE marking

CWR Bit	ECE Bit	Congestion	CWND change
0	1	No congestion	Increase cwnd additively
1	0	Incipient congestion	Decrease multiplicatively
1	1	Moderate Congestion	Decrease multiplicatively
Packet Drop		Severe congestion	Decrease multiplicatively

In TCP header it has the ECN-Echo (ECE) flag and Bit 8 is designated as the Congestion Window Reduced (CWR) flag. These two bits are used both for the initializing phase in which the sender and the receiver negotiate the capability and the desire to use ECN, as well as for the subsequent actions to be taken in case there is congestion experienced in the network during the established state.

When a router has decided from its active queue management mechanism, to drop or mark a packet, it checks the IP-ECT bit in the packet header. It sets the CE bit in the IP header if the IP-ECT bit is set. When such a packet reaches the receiver, the receiver responds by setting the ECN-Echo flag (in the TCP header) in the next outgoing ACK for the flow. The receiver will continue to do this in subsequent ACKs until it receives from the sender an indication that it (the sender) has responded to the congestion notification. Upon receipt of this ACK, the sender triggers its congestion Avoidance algorithm by halving its congestion window, *cwnd*, an updating its congestion window threshold value

1.3. Evaluation of the proposed module

In Adaptive MECN, the objective is to maintain the queue near the *target* queue. If the average queue doesn't vary and remains constant at *target* queue, then the probability of packet drop/mark will remain fixed. Let this probability be *ptarget*.

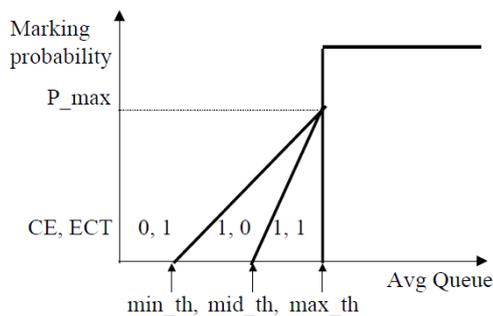


Figure 1. Probabilities of marking packets

We set the *target* queue to be in between *minth* and *midth*. Hence only the first probability curve will be active, in this region. Hence the probability *ptarget* is given by,

$$P_{target} = \frac{P_{max}}{max_{th} - min_{th}} * Averagequeue - min_{th} \quad (1)$$

Since in the above equation, *ptarget*, *minth*, *maxth* are all constant, we can say that,

$$Average\ queue \propto \frac{1}{P_{max}} \quad (2)$$

In any network, we do not have the control over the traffic and the average queue increases or decreases with the load. But the aim is to have the *avgqueue*, always equal to the *targetqueue*. Hence if the *avgqueue*, is greater than *targetqueue*, at any instant, we need to increase *pmax* which would decrease the *avgqueue* so that it becomes equal to *targetqueue* and if the *avgqueue*, is less than, at any instant, we need to decrease *pmax*, to allow the queue, to grow, which would give a better throughput. Thus to keep a constant queue we need to adapt the *pmax*.

Also we need to get the other parameters like *wq*, *maxth*, *midth* and *minth* automatically. Adapt *pmax* in response to measured queue lengths and set *wq*, *maxth*, *midth* and *minth* automatically, based on the link speed and target queue.

2. DESIGN AND IMPLEMENTATION:

Algorithm:-

```

For every (Time Interval) do
  If (avgqueue > ptarget and
  pmax <= Time_Interval) do
    Increase pmax by adding  $\alpha$  to it
  EndIf
  ElseIf (avgqueue < ptarget and pmax >= 0.01)
    Decrease pmax by multiplying  $\beta$  to it
  EndElseIf
End For.

```

The overall Adaptive ECN, which is implemented, has the following features:

Pmax is adapted to keep the average queue size with a target range half way between *minth* and *maxth*

Pmax is adapted slowly, over time scales greater than a typical round-trip time and in small steps. The time scale is generally 5-10 times the typical RTT of the network.

Pmax is constrained to remain with range of [0.01,Time_Interval]

Instead of multiplicatively increasing and decreasing *pmax* here the policy used is additive-increase multiplicative-decrease (AIMD) policy.

The robustness of this algorithm comes from its slow and infrequent adjustment of *pmax*. The price of these slow modifications is that after a sharp change in the level of congestion, it could take some time, before *pmax* adapts to its value. But also adapting α and β makes this process faster and decrease the response time of the system.

2.1 Setting the Parameters:

The range for *pmax*: The upper bound of 0.5 on *pmax* can be justified because, when operating under the gentle mode, this would mean that the packet drop rate varies from 0 to *pmax*, when average queue varies from *minth* to *maxth* and varies from *pmax* to 1.0, if queue changes from *maxth* to 2**maxth*. For scenarios with very small drop rates, MECN will perform fairly robustly with *pmax* set to the lower bound 0.01, and no one is likely to object to an average queue size less than the target range.

Parameters α and β :

The α is an increase factor which can be given as,

$$\alpha = const * \frac{avg - target}{target} * P_{max}$$

And β is a decrease factor which can be given as,

$$\beta = 1 - X * \frac{target - avg}{target}$$

where,

$$X = const * \frac{target}{target - min}$$

Here the *const* varies from 0 to 1. According to networks speed or condition we are going to set its value. It takes $0.49/\alpha$ intervals for *pmax* to increase from 0.01 to 0.5; this is 24.5 seconds, if α is set as 0.01. Similarly, it takes at least $\log 0.02/\beta$ intervals for *pmax* to decrease from 0.5 to 0.01; with the default values, which is 20.1 seconds. Therefore if there is a sharp change in the router load, then it may take as long as 24.5 seconds for the average queue to reach the target range. This time is really a long time in network. Hence we believe that α and β should also be adapted, according to the position of the average queue, with respect to the target queue. So the value of α and β are also recalculated every 0.5 seconds when the *pmax* calculation is done. We scale the value of β from 0.83 to 1.0 when average queue, varies from 0 to target queue.

Thus use the formula given below to adapt β .

$$\beta = 1 - (0.17 * \frac{(target - avg)}{(target - min)}) \quad (3)$$

Setting midth, maxth and wq: To reduce the need for other parameter-tuning, we also give some guidelines for setting the *midth*, *maxth* and *wq*. The *maxth* is set to three times the *minth*. In this case the target average queue size is centered around 2 * *minth*. We believe that, the target queue should be kept in the low congestion region (i.e. between *minth* and *midth*), to maximize the throughput, but at the same time the *midth* should not be too far from the *targetqueue*, so that when the average queue rises above target, a quick response to congestion is achieved, when the second probability curve,

comes into action. This belief, led us to setting the *midth* slightly above the *targetqueue*.

Thus *midth* was set at $2.25 * \text{midth}$ ($\text{targetqueue}=2*\text{minth}$).

If the queue size changes from one value to another it takes $-1/\ln(1-wq)$ packet arrivals for the average queue to reach 63% of the way to the new value. Thus we refer to $-1/\ln(1-wq)$ as the time constant of the estimator for the average queue size. We set *wq* as a function of the link bandwidth. For MECN in automatic mode, we set *wq* to give a time constant for the average queue size estimator of one second.

Thus we set,

$$wq = 1 - \exp\left(-\frac{1}{C}\right) \quad (4)$$

where *C* is the link capacity in packets/second, computed for packets of the specified default size.

3. CONCLUSION

In today's TCP networks, explicit congestion notification (ECN) is the only explicit mechanism which delivers congestion signals to the source. We present a traffic management scheme based on an enhanced ECN mechanism. In particular, we used adaptive ECN, which conveys more accurate feedback information about the network congestion status than the current ECN scheme. We have designed a TCP source reaction that takes advantage of the extra feedback information that have received from receiving end in the form of notification packet and tunes better, its response to the congestion than the current schemes. So to enhance the networks transmission by using the Adaptive ECN technique we attain feasible solutions to avoid the network congestion. The further work is to

implement this technique in real time system and to observe the behavior of the systems.

4. REFERENCES

- [1] Random Early Detection Gateways for Congestion Avoidance Sally Floyd and Van Jacobson, Lawrence Berkeley Laboratory, University of California
- [2] Network Working Group K. Ramakrishna, Request for Comments: 3168 TeraOptic Network.
- [3] ECN protocols and the TCP paradigm, Teunis J. Ott, June 30 1999 [5] Bayesian Packet Loss Detection for TCP mahur Fonseca and Mark Crovella
- [4] An ECN-based end-to-end congestion-control framework: experiments and evaluation Koenraad Laevens Peter Key Derek McAuley
- [5] The Power of Explicit Congestion Notification Aleksandar Kuzmanovi Department of Compute Science Northwestern University
- [6] S. Floyd and V. Jacobson, "Random early detection gateways for congestion avoidance," IEEE/ACM Transactions on Networking, vol. 1 no. 4, pp. 397-413, August 1993
- [7] Promoting the Use of End-to-End Congestion Control in the Internet Sally Floyd and Kevin Fall : IEEE/ACM Transactions on Networking, vol. 7, no. 4, pp. 458-472, 1999. [Online]. Available: citeseer.nj.nec.com/article/floyd99promoting.html
- [8] Adaptive Multi-level Explicit Congestion Notification, Mukundan Sridharan, Arjan Durrezi, Raj Jain Dept. of Computer and Information Science, The Ohio State University, Columbus, OH

- [9] S. Floyd, "Red: Discussions of setting parameters," <http://www.aciri.org/floyd/REDparameters.txt>, November 1997.
- [10] V. Jacobson, K. Nichols, and K. Poduri, "Red in a different light," 1999. [Online]. Available: citeseer.nj.nec.com/jacobson99red.html
- [11] T. Ziegler, S. Fdida, and C. Brandauer, "Stability criteria for red with bulk-data tcp traffic," Technical report, August 2001. [Online]. Available: <http://www-rp.lip6.fr/sf/WebSF/PapersWeb/red.net2000.pdf>
- [12] K. Ramakrishnan and S. Floyd. The addition of explicit congestion notification to IP, Sept. 2001. Internet RFC 3168
- [13] A. Kuzmanovic. The power of explicit congestion notification (extended version). Northwestern University Technical Report, May 2005.
- [14] L. Le, J. Aikat, K. Jeffay, and F. Smith. The effects of active queue management on web Performance. In Proceedings of ACM SIGCOMM '03, Karlsruhe, Germany, Aug. 2003.
- [15] L. Le, J. Aikat, K. Jeffay, and F. Smith. Differential congestion notification: Taming the elephants. In Proceedings of IEEE ICNP '04, Berlin, Germany, Oct. 2004.
- [16] R. Mahajan, S. Floyd, and D. Wetherill. Controlling high-bandwidth flows at the congested router. Proceedings of IEEE ICNP '01, Riverside, CA, Nov. 2001.
- [17] V. Paxson and M. Allman, Computing TCP's retransmission timer, Nov. 2000. Internet RFC 2988.
- [18] Jain, R., "A Delay-Based Approach for Congestion Avoidance in Interconnected Heterogeneous Computer Networks", Computer Communication Review, V.19 N.5, October 1989, pp. 56-71.
- [19] Jain, R., "Congestion Control in Computer Networks: Issues and Trends", IEEE Network, May 1990, pp. 24-30.