No Reference Video Quality Assessment Based on Least Squares Support Vector Machines

Amitesh Kumar Singam and Venkat Raj Reddy Pashike

December 11, 2022

No Reference Video Quality Assessment based on Least Squares Support Vector Machines

1st Amitesh Kumar Singam
IEEE Computational Intelligence Society
Institute of Electrical and Electronics Engineers
Hyderabad, India
0000-0002-2532-0989

2rd Venkat Reddy Raj Pashike
Blekinge tekniska hogskola
venkatreddy.pashike@gmail.com
Telangana, India

***Abstract*—Now a days, need of Application developers towards developing front end based video applications like Skype or others which forced in huge competition between quality of service and experience. Out of all existing approaches, we considered no reference video quality assessment and moreover, our interest lies towards formulating and melding effective features into one model based on human visualizing characteristics. This research explores the trade offs between quality prediction and complexity towards identifying sparseness of LSSVM model and also involves in feature extraction of h.264 bit stream information extracted at macro block layer towards building up of a machine learning based model for quality Assessment. These features which are expected to have high correlation with the perceptual quality of the videos and We concluded that our proposed model outperformed in terms of performance but only in the case of subjective quality assessment and more over due to refining process of subjective scores, fault in encoding process was traced out which is based on error concealment and in case of building up of proposed model with SSIM and MS-SSIM metric at frame level sparseness was traced out.**

I. Introduction

Generally the evaluation of video quality is classified into two methods. They are subjective and objective analysis of video quality. Subjective analysis is conducted based on human perception since it is concerned with how video is perceived by a viewer or subject and expresses subjects opinion on a particular video sequence in comparison with its original video sequence. The subject has to vote for the video sequences under certain test environment conditions for example the ITU-Recommendations. Human perception is considered as the true judgment of video quality and precise measurement of perceptual quality but it is quite expensive and tedious in terms of time such as preparation, running and human resources. Objective quality assessment is therefore essential. Objective Video Quality Metric should be designed based on HVS (Human Visualizing System) characteristics. Some aspects of HVS like contrast, orientation sensitivity, spatial and temporal masking effects, frequency selectivity and color perception are incorporated in the design of objective quality metrics. Even though it is computationally very expensive and complex to design a quality metric with above aspects. It is useful for a wide range of applications if it correlates well with human perception.

a) : The impairments visibility which is related to video processing system is subjected to spatial and temporal properties of video content, since subjective analysis is quite expensive and time conservative method, objective metric has been developed considering HVS. In our thesis work we have done experiment on both subjective and objective analysis.

II. Subjective Video Quality Assessment

Results of subjective assessment largely dependents on the factors like selection of test video sequences and welly defined evaluation procedure. In our research work, we carefully employed the specifications recommended by ITU-R BT 500-10 [1]as mentioned and VQEG Hybrid test plan in which is explained briefly in following sections.

A. Test Video Sequences

In this paper six different video sequences of CIF and QCIF spatial resolutions were selected in raw progressive format based on different motion content and including various levels of spatial-temporal complexity recommended by ITU-R P.910. The measurement of spatial-temporal information is essential due to quality of transmitted video sequence is highly dependent on this whereabouts. Each of 120 video sequences are of 10 seconds long with fast, medium and slow motion content. In this paper the generation, scaling and encoding process of Test videos sequences were done using JM reference software 16.1 based on H.264 standards. In Group Of Pictures, intra coded frame contains high quality but gradually reduced for following sequence of P-frames and the occurrence of I frame effect will have huge influence on subjective test. Since the I-frame plays important role in testing video sequences, It has been finalized not to consider initial 10 seconds for quality measurement. With the help of virtual dub, same video sequence was doubled up from 10sec to 20sec without changing temporal and spatial resolutions. Finally after encoding process, all test sequences were cropped last 10 out of 20 seconds long in order to overcome I frame effect.

Out of 120 videos, the video sequences encoded at 15fps are temporally up scaled to 30fps by repeat frame method and QCIF videos are spatially up scaled to CIF using Bi-cubic Interpolation method with the help of Virtual dub. Finally, subjective analysis was conducted for 120 test sequences at temporal resolution of 30fps and spatial resolution of 352x288(CIF).

B. Methodology for Subjective Experiment

Subjective experiment was conducted, even though It is costlier and time consuming method. Subjective scores are obtained by evaluation of video quality with involvement of human observers by grading them according to his/her perception. Subject will grade the visual quality of test sequences by in the form of Mean Opinion Score, therefore average of overall scores results in obtaining subjective measure of video quality. This experiment was conducted under laboratory viewing environment specified by ITU-R BT.500-12 Standards. Single Stimulus Continuous Quality Evaluation(SSCQ) process was selected out of Single Stimulus and Stimulus Comparison Quality Evaluation. SSCQ method considers hidden reference which leads in obtaining efficient results. The grading values given for hidden reference are only used to consider the seriousness of the subject but not included in final results. In this method subjects will be able to observe video sequence once and grade it in the given 10 seconds time based on his/her perception and All Test sequences have been played in pseudo random order. Each video sequence was displayed for 10 seconds with grading scale of 0-100. Raw scores distribution before refining.

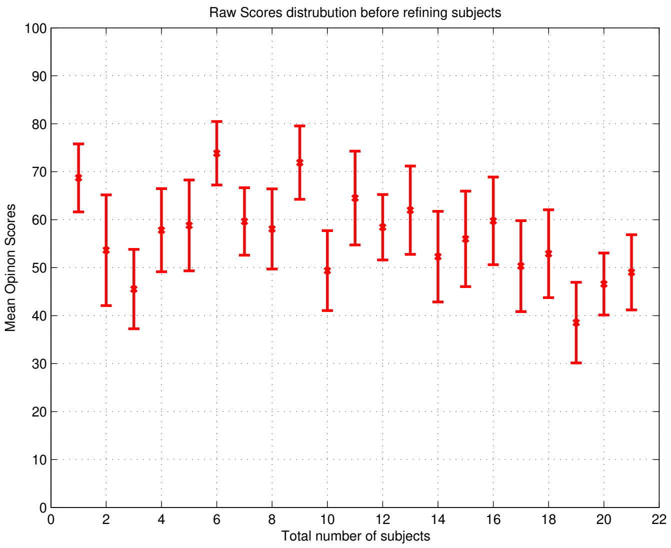

Fig. 1: Raw scores distribution before refining

C. Data screening for SSCQE method

In order to obtain significant results out of raw subjective scores, two step filtering method was employed. The first step is keen to detect and discard the observers those who exhibited great change of votes compared to average behavior. The function of second step is to detect and reject the screening of inconsistent observer without any thought of systematic change.

1) Step 1: For confirming that obtained scores for each time window of test configuration is normal distribution or not, β_2 test was conducted and Finally for achieving fist step of refining raw scores, mean $\bar{u}_{jklr}$, standard deviation S_{jklr} and the coefficient β_{2jklr} for each of all the time windows of each test configuration was computed.

$$\beta_{2jklr} = \frac{m_4}{(m_2)^2}. \tag{1}$$

where

$$m_x = \frac{1}{N} \sum_{n=1}^{N} (u_{njklr} - \bar{u}_{jklr})^x.$$

$$\bar{u}_{jklr} = \frac{1}{N} \sum_{n=1}^{N} u_{njklr}.$$

- n is number of observers.
- j is number of time windows within combined test sequence and condition.
- k is number of test conditions.
- l is number of test sequences
- r is number of repetitions.

P_i and Q_i for i^{th} observer were calculated. Where P_i and Q_i are maximum and minimum scores of test sequences given by i^{th} observer.

if($2 \leq \beta_{2jklr} \leq 4$)then:

$$\text{if } u_{njklr} \geq \bar{u}_{jklr} + 2S_{jklr} \text{ then } P_i = P_i + 1$$

$$\text{if } u_{njklr} \leq \bar{u}_{jklr} + 2S_{jklr} \text{ then } Q_i = Q_i + 1$$

else

$$\text{if } u_{njklr} \geq \bar{u}_{jklr} + \sqrt{20}S_{jklr} \text{ then } P_i = P_i + 1$$

$$\text{if } u_{njklr} \leq \bar{u}_{jklr} - \sqrt{20}S_{jklr} \text{ then } Q_i = Q_i + 1$$

where

$$S_{jkr} = \sqrt{\sum_{i=1}^{N} \frac{(\bar{u}_{jklr} - u_{ijklr})^2}{N-1}}$$

$$\text{if } \frac{P_i}{J.K.L.R} \geq 0.2 \text{ or } \frac{Q_i}{J.K.L.R} \geq 0.2 \text{ then reject observer i}$$

The above process reject observers based on scores significantly distant from average scores produced by subjects.

2) step 2: this method actually detects and discards the observers based on consistency of votes given and similarly, the distribution of scores are normal or not is confirmed by the means of β_2 test. To achieve final step of refining raw scores, mean $\bar{u}^*_{jklr}$, standard deviation S^*_{jklr} and the coefficient β^*_{2jklr} for each of the time windows of each test configuration are calculated.

$$\beta^*_{2jklr} = \frac{m_4}{(m_2)^2}. \tag{2}$$

where

$$m_x = \frac{1}{N} . \sum_{n=1}^{N} (u^*_{njklr})^x.$$

The centered scores $u*_{njklr}$ are computed as follows

$$u*_{njklr} = u_{njklr} - u_{nklr} + \bar{u}_{klr}.$$

The mean score for each test configuration is computed as

$$\bar{u}_{klr} = \frac{1}{N.J} . \sum_{n=1}^{N} \sum_{j=1}^{J} u_{njklr}.$$

u_{njklr} is score of i^{th} observer for j^{th} time window and k^{th} test condition for l video sequences with repetition r. The mean score for each observer and for each test configuration is computed as

$$\bar{u}_{nklr} = \frac{1}{J}\sum_{j=1}^{J} u_{njklr}.$$

we need to calculate P_i^* and Q_i^*, for i^{th} observer and where P_i^* and Q_i^* are maximum and minimum scores of test sequences given by i^{th} observer.

if$(2 \leq \beta_{2jklr} \leq 4)$then:

$$\text{if } u_{njklr}^* \geq \bar{u}_{jklr}^* + 2S_{jklr}^* \text{ then } P_i^* = P_i^* + 1$$

$$\text{if } u_{njklr} \leq \bar{u}_{jklr}^* - 2S_{jklr}^* \text{ then } Q_i^* = Q_i^* + 1$$

else

$$\text{if } u_{njklr} \geq \bar{u}_{jklr}^* + \sqrt{20}S_{jklr}^* \text{ then } P_i^* = P_i^* + 1$$

$$\text{if } u_{njklr} \leq \bar{u}_{jklr}^* - \sqrt{20}S_{jklr}^* \text{ then } Q_i^* = Q_i^* + 1$$

$$\text{if } \frac{P_i^*+Q_i^*}{J.K.L.R} \geq 0.1 \text{ or } \frac{P_i^*-Q_i^*}{P_i^*+Q_i^*} \geq 0.3 \text{ then reject observer i}$$

Raw scores distribution after refining.

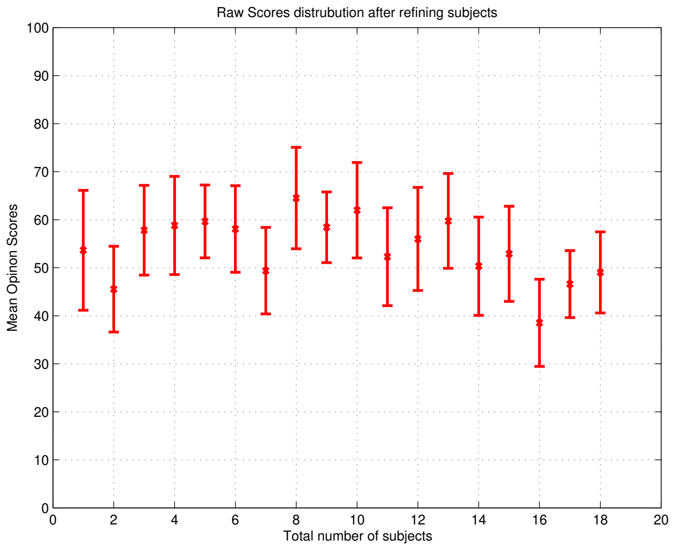

Fig. 2: Raw scores distribution after refining

III. Objective Video Quality Assessment

The most commonly used quality metrics for predicting video quality,are PEVQ, PSNR, MSE and SSIM. Though PSNR and MSE are pixel by pixel based metric, because of some issue regarding quality assessment resulted in development of other metrics like Structural Similarity Index Metric(SSIM), Opticoms PEVQ. Peak signal to noise ratio(PSNR) is expressed as

$$PSNR = 10\log\frac{MAX_t{}^2}{MSE(n)}$$

MAX_t is maximum pixel value and MSE is average of square of difference between luminance values of corresponding pixels between two frames.

$$MSE = \frac{1}{UV}\sum_{u=1}^{U}\sum_{v=1}^{V}[I_R(u,v) - I_D(u,v)]^2 \tag{3}$$

$I_R(u,v)$ is intensity value of reference frame at pixel location (u, v) and$I_D(u,v)$is intensity Value of distorted frame at pixel location (u, v). U and V are number of rows and columns in a video frame. PSNR is calculated for entire sequence of video of length N is expressed as

$$PSNR = \frac{1}{N}\sum_{n=1}^{N} PSNR(n) \tag{4}$$

Structural Similarity Index is a quality metric which measures the structural similarity between two frames. SSIM is a still used as an alternative for evaluation of perceptual video quality. SSIM considers quality degradations in the frames as perceived changes in the variation of structural information between frames of distorted and original video sequences.

$$SSIM(n) = \frac{[2\mu_{I_R}(n)\mu_{I_D}(n)+C_1][2\sigma_{I_R I_D}(n)+C_2]}{[\mu_{I_R}{}^2(n)+\mu_{I_D}{}^2(n)+C_1][\sigma_{I_R}{}^2(n)+\sigma_{I_D}{}^2+C_2]}$$

$\mu_{(I_R)}(n), \mu_{(I_D)}(n)$ are mean intensity of n^{th} frame of reference video (I_R) and distorted video (I_D), $\sigma_{(I_R)}(n)$ and $\sigma_{(I_D)}(n)$ are contrast of n^{th} frame of reference video (I_R) and distorted video (I_D). C_1, C_2 are constants used in order to evade any instabilities in the structural similarity comparison. SSIM is calculated for entire sequence of video of length N

$$SSIM = \frac{1}{N}\sum_{n=1}^{N} SSIM(n) \tag{5}$$

Multi-scale structural similarity(MS-SSIM) approach provides further flexibility than previous methods in integrating the variations of conditions like display resolution and viewing distance. MS-SSIM actually calibrate the factors that states the relative importance of different scales.

$$MS{-}SSIM(x,y,n) = [l_M(x,y)]^{\alpha_M}\prod_{j=1}^{M}[c_j(x,y)]^{\beta_j}[s_j(x,y)]^{\gamma_j} \tag{6}$$

$c_j(x,y)$ and $s_j(x,y)$ denotes calculation of Contrast and Structure Comparison at j^{th} scale. $l_m(x,y)$ denotes computation of luminance comparison only at scale M. MS-SSIM is calculated for entire sequence of video of length N

$$MSSIM = \frac{1}{N}\sum_{n=1}^{N} MSSIM(x,y,n) \tag{7}$$

PEVQ measures quality of video based on mean square of two frames for luminance component. PEVQ has been developed for low resolutions such as CIF and QCIF in which motion information is used for forming the final measure. PEVQ is a standardized end-to-end measurement algorithm which estimates mean opinion scores of the video quality by modeling the behavior of the human visual system and it has become a part of ITU-T Recommendation J.247.

IV. Feature Extraction of H.264 Bitstream Data

The feature extraction process for H.264 coded bitstream data was performed in two main steps. First the encoded video bitstreams were decoded using a modified version of JM reference software 16.1 in order to generate an XML file of coding parameters for each video sequence. These XML files contained video information at macroblock level such as quantization parameters, absolute and difference motion vectors, and the type of macroblocks. In the next step a Java program was developed to analyze and process the large XML data in order to extract 18 selected features of the coded videos at frame level. These features which are expected to have high correlation with the perceptual quality of the videos

V. Kernel Based Learning Concept

Kernel based learning methods are classified into supervised and unsupervised learning algorithms. Kernel method solves any problem by mapping the input data set into high dimensional feature space via linear or nonlinear mapping which is also referred as kernel trick. In recent years, few powerful kernel based models were proposed such as support vector machines, kernel fisher discriminant and kernel principal component analysis which are used for regression, classification, dimensionality reduction and other jobs. In our research work, we adapted Support Vector Machines(SVM) algorithm for regression analysis.

A. Support vector Machines

Support Vector machine(SVM) is a supervised and powerful learning Based algorithm invented by Vladimir VaDnik [2] and it is commonly used for classification and regression analysis. Its formulation is based on structural risk minimization principle which includes capacity control in order to prevent over-fitting problem of Empirical Risk Minimization principle based learning algorithms like traditional Neural Networks. In our research work, we performed regression analysis where Support Vector Regression exhibits the benefits of machine learning with the capability of learning difficult data patterns by mapping of simplified input features extracted from h.264 bit stream data and regressing with desire or true values in a very effective way. The mechanism of SVM works by mapping of nonlinear input data to high dimensional kernel induce space via nonlinear mapping which leads to solving set of linear equations in kernel space. An insensitive loss function is introduced in SVM which measures the risks and the kernel functions has the flexibility that allows SVM to search a wide variety of hypothesis spaces. In order to overcome inequality constraints Suykens et al. [3] developed variant of Support Vector Machine, Least Squares Support Vector machines(LSSVM) reformulates the standard SVM which leads to solve linear Karush-Kuhn-Tucker systems. LSSVM formulations reduce computational calculations of standard SVM. LSSVM transforms quadratic programming into a set of linear equations which is easier than solving QP problems.

1) Least Square Support Vector Regression:

Given set of training data points $[y_i, z_i]_{i=1}^{d}$.

since $y_i \epsilon R^m$ is m-dimensional feature vector and d is dimension of training set and the output is $z_i \epsilon R$. In LS-SVM, a linear estimation is obtained in a kernel-induced feature space by following

$$Z(y) = \omega^T \phi(y) + b \tag{8}$$

Where $\phi(y) : R^m \rightarrow R^{n_d}$, the weight vector is $\omega \epsilon R^{n_d}$ is primal weight space. $\phi(y)$ represents high dimensional kernel induced feature space were input data are mapped via nonlinear mapping and it is referred as kernel trick.
Similar to standard SVM, optimization problem in lssvm is formulated for prediction function as following.

$$Min_{\omega,e} J(\omega, e) = \frac{1}{2}\omega^T \omega + \frac{1}{2}\gamma \sum_{i=1}^{d} e_{i^2} \tag{9}$$

$$Z_i = \omega^T \phi(y_i) + b + e_i \tag{10}$$

Where i=1, 2,...d. And error variable is $e_i \epsilon R$, b denotes bias term. From Standard SVM which has a typical convex optimization problem that can be solved with the help of Lagrangian multipliers method. Lagrangian is defined by

$$L(\omega, b, e, \alpha) = J(\omega, e) - \sum_{i=1}^{d} e_i[\omega^T \phi(y_i) + b + e_i - Z_i] \tag{11}$$

Where lagrangian multiplier $\alpha_i \epsilon R$, The conditions for optimality are given by

$$\begin{cases} \frac{\partial L}{\partial \omega} = 0 \rightarrow \omega = \sum_{i=1}^{d} \alpha_i[\phi(y_i)] \\ \frac{\partial L}{\partial b} \rightarrow \sum_{i=1}^{d} \alpha_i[\phi(y_i)] \\ \frac{\partial L}{\partial e_i} \rightarrow \alpha_i = \gamma e_i, i = 1, 2..., d. \\ \frac{\partial L}{\partial \alpha_i} \rightarrow Z_i = \omega^T \phi(y_i) + b + e_i, i = 1, ...d. \end{cases} \tag{12}$$

Excluding $a_i = \gamma e_i$, the above conditions which are similar to SVM optimal conditions, obtained solution is derived as follow

$$\sum_{i=1}^{d} \alpha_i \phi^T(y_i)\phi(y_j) + b + \alpha_j \frac{\alpha_j}{\gamma} = Z_j \tag{13}$$

$$K_{ij} = K(y_i, y_j) = \phi^T(y_i)\phi(y_j) \tag{14}$$

Since $y_i, y_j \epsilon$, i,j=1,.....,d. After the eliminating ω, e_i the solution is we obtained set of following linear equations.

$$\begin{bmatrix} 0 & l^T \\ 1 & K + \Delta \end{bmatrix} \begin{bmatrix} b \\ \alpha \end{bmatrix} = \begin{bmatrix} 0 \\ Z \end{bmatrix} \tag{15}$$

where $Z = (Z_1, ...Z_d)^T$, $Z = (k + \Delta)\alpha$+b1
Δ=diag($\frac{1}{\gamma}, ...\frac{1}{\gamma}$), $\alpha = (\alpha_1, ..., \alpha_d)^T$
$\alpha_1 + \alpha_2 + ...\alpha_d = l^T\alpha$, $l = (l_1, ...l_d)^T$.
The prediction function of LS-SVM model is derived from as

$$Z(x) = b + \sum_{i=1}^{d} \alpha_i K(y, y_i) \tag{16}$$

Where K(..,..)defines kernel function. We selected RBF kernel function which is well suited for our application domain and is expressed as

$$K(y, y_i) = exp\frac{-\,||y - y_i||\,|^2}{\sigma^2} \tag{17}$$

where σ is width or insensitive zone of RBF kernel.

2) Test Methodology of LSSVM Model: RBF was selected as kernel function for realization implicit mapping of give input data into higher dimensional feature kernel space and it provides good performance which results in obtaining better training and testing errors. An optimization algorithm was used for tuning the hyper parameters σ and γ with respect to good performance measure. Grid search method was employed which performs an exhaustive search through a subset of parameter space in machine learning algorithm for solving model selection problem by finding optimal parameters. This algorithm is guided by performance metric(MSE) and measured by leave one out cross validation in training set. The performance and accuracy of LSSVM model depends on (σ^2, γ), Where σ^2 width of kernel and γ is regularization parameter. For each pair of hyper parameters (σ^2, γ) leave one out cross validation method on training set is performed to estimate the prediction error. Therefore a robust model is obtained by selecting those optimal pair of hyper parameters which gives the lowest MSE.

VI. Cross Validation Strategy

In order to avoid the over fitting problem, we need to measure the generalization performance of LSSVM model when data approximation is not used during training process. In this paper, we selected leave one out cross validation technique to obtain better generalization ability. k-fold cross-validation has capability of not only minimizing training error but also reduces cross validation error. Cross-validation is a resampling strategy used to validate the performance of model by random sub-sampling of the available data. In other words, the original data is subdivided randomly into k folds. In each round of CV, k-1 folds are used for training and the remaining one fold is used for validation, this procedure is repeated for k times with each of the k folds should use exactly once for the model validation. Performance estimation of our proposed model is determined by average of results obtained in k folds for k rounds

A. Leave One Out Cross Validation

We employed leave one out cross validation technique for estimating the performance of predictive model, leave one out cross validation method is functions similar to K-fold CV, where K is equal to total number of data points. In each round of LOOCV, single data point is used for validation while remaining data points are used for training and this procedure is repeated such that each of the all data points should involve once for the model validation.

Given data set $D = [(x_1, y_1)..........(y_n, y_n)]$

Training set $T = [(x'_1, y'_1)...........(x'_n, y'_n)]$

For Mϵ (1..c), where M is prediction model, (X, Y) is random variable distributed according to given probability distribution function in order to reduce the Mean square error.

$$\varepsilon_m(i) = E_L(Y, f_m(X)) \tag{18}$$

where E_L is expected loss.

Working methodology of LOOCV is explained in following steps

- Step1: randomly permute the data from given data set
- Step2: Split into N-Folds, each fold is equal to one instance of given input data.
- Step3: for each round use N-1 instances for training and remaining for Validating.

$$MSE_{LOOCV} = \frac{1}{N}\sum_{n=1}^{N}\varepsilon_m(n) \tag{19}$$

  Where ε_m Average of empirical error and $\varepsilon_m(n)$ is empirical error in validation set in each of N rounds for N-instances.
- Step 4: we choose M^* to minimize ε_m and retrain using M^* on all of data and then we get final prediction function

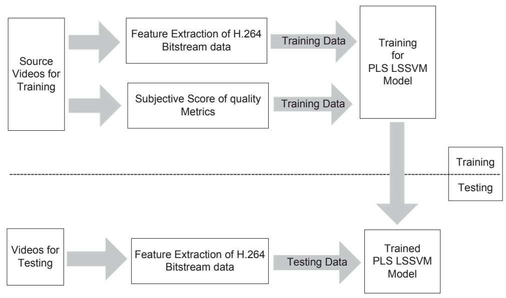

Fig. 3: Architecture Design of Proposed Model for Video quality

VII. Frame Level Quality Evaluation of Proposed Model

For instance, its not per video, we evaluated our proposed model at frame level, Moreover, these features were extracted out of H.264 bit stream data and also These features are expected to have high correlation with corresponding perceptual quality score of the selected video, mainly feature extraction of 120 videos has been processed using JM Reference software Version 16.0. The properties of encoded videos are acquired from bit stream data of H.264/AVC which has been generated as a trace file while encoding process. Rather than using completely decoded frame, our interest lies in reversing the entropy encoding of bit stream. By analyzing three successive Nal, Slice and Macro Block Layer and the following features were extracted.

- Avg QP- Average Quantization Parameter.
- Avg bitrate [kbps]-Average Bits per second
- Inter[%]-Percentage of Inter Coded Macro Blocks
- Intra[%]-Percentage of Intra Coded Macro Blocks

- Skip[%]-Percentage of Skip Coded Macro Blocks
- P16x16[%]-Percentage of Inter Coded Macro Blocks with 16x16 subdivision
- P8x8[%]-Percentage of Inter Coded Macro Blocks with 8x8 subdivision
- P4x4[%]-Percentage of Inter Coded Macro Blocks with 4x4 subdivision
- MV_X- Average of Horizontal absolute Motion Displacement
- MV_Y- Average of Vertical absolute Motion Displacement
- MVD_X- Average of the Motion Displacement difference in horizontal direction

$$MVD_X = |MV_x(i_l.j) - MV_x(i_r, j)| \tag{20}$$

Where (i_l, j) and (i_r, j) positions at left and right edge image or frame

- MVD_Y- Average of the Motion Displacement difference in vertical direction

$$MVD_Y = |MV_y(i_l.j) - MV_y(i_r, j)| \tag{21}$$

Where (i_l, j) and (i_r, j) positions at left and right edge image or frame

- Zero MVs[%]-Percentage of Zero absolute Motion vectors
- Zero MVDs[%]-Percentage of Zero Motion vector Difference
- Motion Intensity

$$MI1 = \sum_{i=0}^{N} \sqrt{MV_{X^2{}_i} - MV_{Y^2{}_i}} \tag{22}$$

where N is the total number of macro blocks in each frame. MV_{X_i} and MV_{Y_i} are the absolute motion vector of the i-th macro block in Horizontal(X) and Vertical(Y) directions respectively.

- Motion Intensity II

$$MI2 = \sqrt{MV_{X^2} - MV_{Y^2}} \tag{23}$$

where MV_X and MV_Y are the average of absolute motion vectors in each frame in X and Y directions respectively.

- IinPframes[%]-Percentage of Intra coded macro blocks in P frames.

TABLE I: Statistical results of proposed model

VQM	MOS	PEVQ	SSIM	PSNR	MSSIM
PCC	0.96	0.98	0.99	0.94	0.94
SROC	0.95	0.97	0.98	0.93	0.94
OR	0.96	0.98	0.98	0.93	0.94
MSE	0.94	0.97	0.98	0.93	0.94

Below figure illustrates correlation between predicted and target scores of Ms-SSIM.

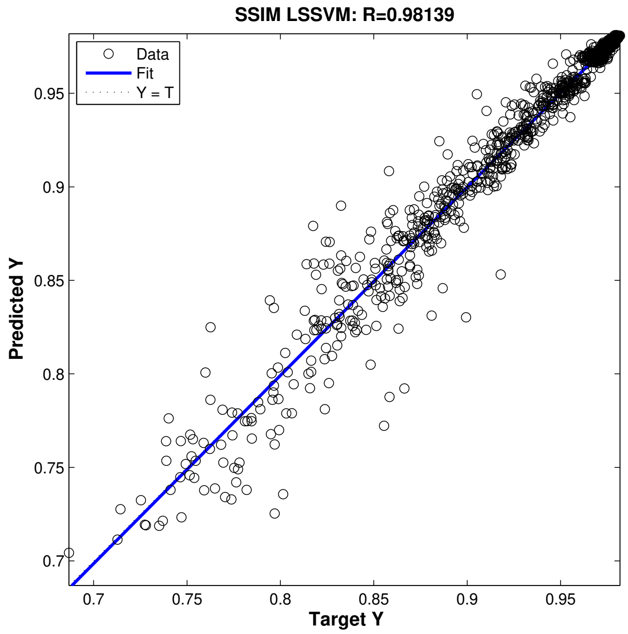

Fig. 4

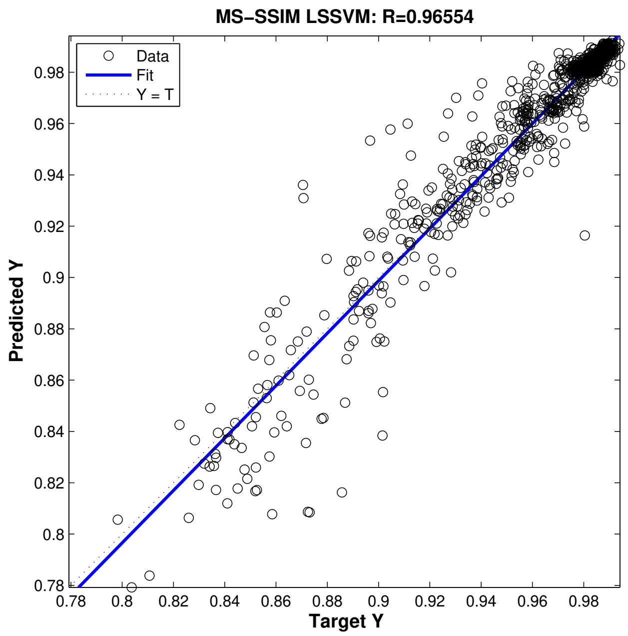

Fig. 5

Below figure illustrates correlation between predicted and target scores of SSIM.

VIII. Conclusion

We concluded that our proposed model outperformed in terms of performance but only in the case of subjective quality assessment and more over due to refining process of subjective scores, fault in encoding process was traced out based on error concealment and in case of building up of proposed model with SSIM and MS-SSIM at frame level, we identified sparseness and our future work is based on eliminating sparseness of proposed model.

REFERENCES

[1] "ITU-R radio communication sector of itu, recommendation itu-r bt.500-12," 2009, http://www.itu.int/.
[2] Vapnik.V, *Statistical learning theory*. Wiley, 1998.
[3] S. J. A. K and Vandewalle.J, "Least squares support vector machine classifiers [j]," 1999, neural Processing Letter.