IJCRT.ORG ISSN : 2320-2882

INTERNATIONAL JOURNAL OF CREATIVE RESEARCH THOUGHTS (IJCRT)

An International Open Access, Peer-reviewed, Refereed Journal

Analysis and Simulation of H.264 Decoder AVC

, [1]Dr. Prasantha H S, [2] M V Sai Thanmai, [3]Likitha G, [4] Kiran, [5] G Jeevana

[1]Professor, Dept. of Electronics and Communication Engineering

[2,3,4,5] Undergraduate Student, Dept. of Electronics and Communication Engineering

Nitte Meenakshi Institute of Technology, Bangalore, India

Abstract: As technology advances, multimedia applications increase exponentially in day-to day life. Multimedia applications such as video telephony, video conferencing, TV, streaming video/audio online, and many other applications are in demand in video industry. These applications usually require high bandwidth, large storage, and high latency time to send on network. To conserve resources, it is required to compress the video data before sending them to the network by sender side. It is also required to decompress the video data at receiver end before broadcasting. Many different video codec standards such as H.261, MPEG-1, MPEG-2, H.263, and H.264 are implemented. H.264 is latest international video codec standard. This protocol was developed jointly by International Telecommunication Union – Telecommunication Standardization Sector (ITU-T) and International Organization for Standardization (ISO).

The objective of this work is to explore different blocks of H.264 in MATLAB environment. This project first briefly describes about decoding process, and then it discusses more details about different modules of decoder, and the related algorithms. Finally, it gives result based on compression ratio, peak signal to noise ratio. A video file is given as input to encoder, the video file is converted to a number of frames using video codec, and fixed size macro block is defined in each frame for encoding process. Motion search algorithm finds motion vector after macro block definition, then compensated image is generated based on reference frame and motion vector by video codec. Redundancy is removed from current frame by subtracting compensated image. Compression of residual information is performed using transformation, quantization, and entropy coder. Compressed data are given as inputs to decoder, and decoder process the image to reconstruct image. To enhance image quality and reduce blocking artifact, the image frame is passed through filter. The project is completed successfully by reconstructing video with reasonable quality.

Index Terms **– Video Processing, Video Coding, H.264, Decoder**

I. INTRODUCTION

H.264 is widely accepted video codec standard. It is also known as MPEG-4 Part 10 Advance Video Codec (AVC). Some of its features are high-definition resolution, interlaced/progressive scan mode, variable frame rate, high bit rate, supporting I, B, & P-frames, 9 different prediction modes, variable size block matching motion estimation, and motion compensation, and image enhancement filter. H.264 is compatible with almost all kinds of recent video coding tool as opposed to previous standards. As a result, H.264 is most popular nowadays. Advanced Video Coding also known as H.264 or MPEG 4 part 10. H.264/AVC is the latest in a series of standards published by the ITU and ISO in 2003. It describes and defines a method of coding video that can give better performance than any of the preceding standards. H.264 makes it possible to compress video into a smaller space, which means that a compressed video clip takes up less transmission bandwidth and/or less storage space compared to older codecs. Video coding is the process of compressing and decompressing a digital video signal.

MPEG STANDARDS AND FEATURES:

The main purpose of MPEG is to take analog/digital video signals as input. Convert the input into packets of digital information such that they are efficient to be transported on modern networks. The MPEG compression will help data to consume less bandwidth compare to traditional transmission.

MPEG-1: MPEG-1 is a standard for lossy compression of video and audio. It is designed to compress VHS-quality raw digital video and CD audio down to about 1.5 Mbit/s (26:1 and 6:1 compression ratio respectively) without excessive quality loss, making video CDs, digital cable/satellite TV and digital audio broadcasting (DAB).

Drawbacks of MPEG1:

- The audio compression is limited to two channels.
- There is no standardized support for interlaced video with poor compression when used for interlaced video
- It has a limited standardized profile
- It supports only one-color space - 4:2:0.

MPEG-2: MPEG-2 (a.k.a. H.222/H.262 as defined by the ITU) is a standard for "the generic coding of moving pictures and associated audio information". It describes a combination of lossy video compression and lossy audio data compression methods, which permit storage and transmission of movies using currently available storage media and transmission bandwidth. While MPEG-2 is not as efficient as newer standards such as H.264/AVC and H.265/HEVC, backwards compatibility with existing hardware and software means it is still widely used, for example in over-the-air digital television broadcasting and in the DVD-Video standard.

MPEG 3: It was designed for high-definition television. It was abounded and incorporated to MPEG-2.

MPEG-4: MPEG-4 is a method of defining compression of audio and visual (AV) digital data. It was introduced in late 1998 and designated a standard for a group of audio and video coding formats and related technology agreed upon by the ISO/IEC Moving Picture Experts Group Coding of audio-visual objects.

MPEG-7: MPEG-7 is formally called Multimedia Content Description Interface. It uses XML to store metadata, and can be attached to time-code in order to tag particular events, or synchronize lyrics to a song.

REVOLUTION OF H.26X STANDARDS:

H.261: H.261 is an algorithm that determines how to encode and compress the data electronically. It is a video coding standard published by the ITU (International Telecommunication Union) in 1990. H.261 encoding technique can encode only video part of an audiovisual service. H.261 is a two-way communication over ISDN lines (Video conferencing and Video calling) and supports data rate in multiples of 64 KBPS. H.261 defines a video encoder that is intended to be used to compress video data that will be sent over Integrated Services Digital Network (ISDN) lines. The H.261 codec is intended primarily for use in video telephony and videoconferencing applications.

H.263: The requirements of H.263 standardization was:

- Use of available technology
- Interoperability between the other standards, like H.261Flexibility for future extension
- Quality of service parameters, such as resolution, delay, frame-rate etc
- Subjective quality measurements.

II. LITERATURE SURVEY

This section consists of summary of the survey conducted.

1. Fast Mode Decision for H.264/AVC Based on Rate-Distortion Clustering: Author: Yu-Huan Sung and Jia-Ching Wang, Senior Member, IEEE

This work presents a multi-phase classification (MPC) scheme that builds a mode decision tree according to the clustering of rate-distortion costs. A nearest cluster mean criterion is used to examine candidate modes phase by phase, and a performance control mechanism is incorporated to maintain coding performance. Based on sparse and redundant representations and the K-SVD dictionary learning algorithm they produce good results when compared to other compression technique. This method doesn't work well in fingerprint images.

2. Fast mode decision scheme using sum of the absolute difference-based Bayesian model for the H.264/AVC video standard
   Author: Jongho Kim1, Jong-Hyeok Lee2, Byung-Gyu Kim, Jin Soo Choi

The sum of the absolute difference (SAD) value, including the motion cost of each mode, is used as a classification feature to divide the block modes into several groups. A refinement method using a Bayesian model based on the average SAD value is also proposed. The proposed compression scheme which uses learned dictionaries (preferable learned with RLSDLA) performs quite well. The results produced by this method are comparable but slightly worse than JPEG2000.

3. A Long-Term Reference Frame for Hierarchical B-Picture-Based Video Coding: Author: Manoranjan Paul, Senior Member, IEEE, Weisi Lin, Senior Member, IEEE, Chiew Tong Lau, Member, IEEE, and Bu-Sung Lee, Member
There are two approaches to generate McFIS under the proposed methodology. In the first approach, we generate a McFIS using a number of original frames of a scene in a video and then encode it as an I-frame with higher quality. For the rest of the scene this generated I frame is used as an LTR frame. More number of data can be compressed. Due to the unavailability of the adjacent reference frames or with no explicit background referencing being used, the HBP scheme cannot take full advantage of the MRFs (i.e., more than 2 reference frames) benefits.

4. Rotated orthogonal transform (ROT) for motion-compensation residual coding: Author: Z Gu, W Lin, BS Lee, CT Lau

Discrete cosine transform (DCT) is the orthogonal transform that is most commonly used in image and video compression. The motion-compensation residual (MC-residual) is also compressed with the DCT in most video codecs. However, the MC-residual has different characteristics from a nature image. The process achieves better result when compared with the other methods due to the transformation. Due the dual process like video and motion compensation, the time will be more for encoding and decoding

III. PROPOSED WORK

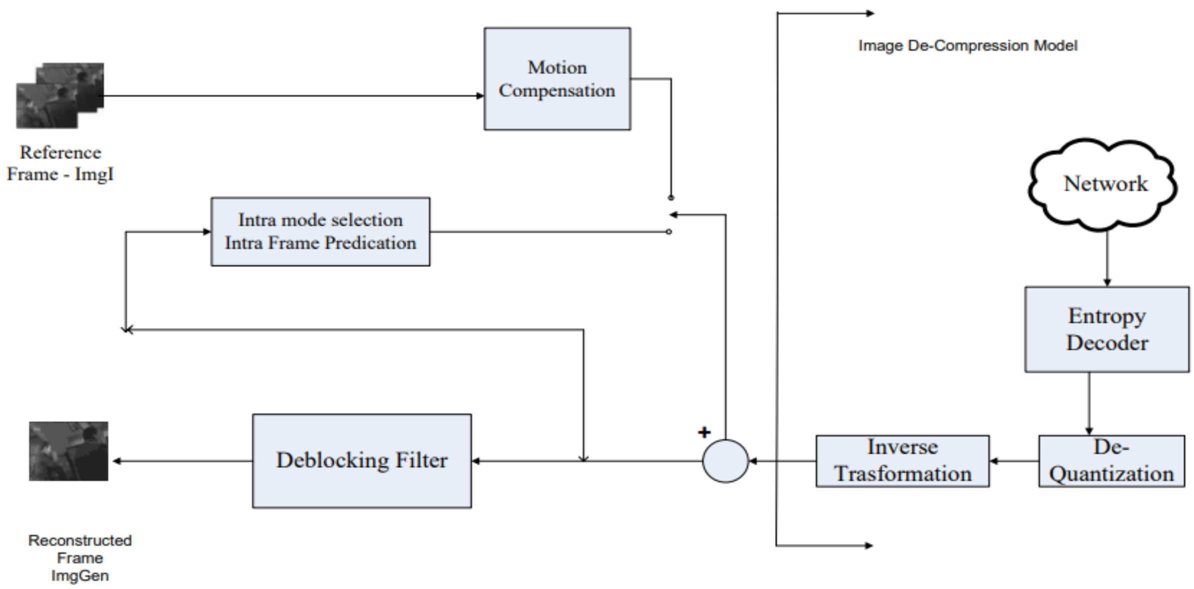

Figure 1: Block Diagram of H.264 Decoder

Frame/picture/block types:

- **Intra coded(I) frame:**
  - I-frames contain an entire image. They are coded without reference to any other frame except (parts of) themselves.
  - May be generated by an encoder to create a random access point (to allow a decoder to start decoding properly from scratch at that picture location).
  - May also be generated when differentiating image details prohibit generation of effective P or B-frames.
  - Typically require more bits to encode than other frame types.

- **Predicted (P) frame:**
  - Require the prior decoding of some other picture(s) in order to be decoded.
  - May contain both image data and motion vector displacements and combinations of the two. Can reference previous pictures in decoding order.
  - Older standard designs use only one previously decoded picture as a reference during decoding, and require that picture to also precede the P picture in display order.
  - In H.264, can use multiple previously decoded pictures as references during decoding, and can have any arbitrary display-order relationship relative to the picture(s) used for its prediction.
  - Typically require fewer bits for encoding than I picture do.

- **Bidirectional predicted (B) frames:**
  - Require the prior decoding of subsequent frame(s) to be displayed.
  - May contain image data and/or motion vector displacements. Older standards allow only a single global motion compensation vector for the entire frame or a single motion compensation vector per macroblock.
  - Include some prediction modes that form a prediction of a motion region (e.g., a macro block or a smaller area) by averaging the predictions obtained using two different previously decoded reference regions. Some standards allow two motion compensation vectors per macro block (biprediction).
  - In older standards (such as MPEG-2), B-frames are never used as references for the prediction of other pictures. As a result, a lower quality encoding (requiring less space) can be used for such B-frames because the loss of detail will not harm the prediction quality for subsequent pictures
  - H.264 relaxes this restriction, and allows B-frames to be used as references for the decoding of other frames at the encoder's discretion.
  - H.264 allows for one, two, or more than two previously decoded pictures as references during decoding, and can have any arbitrary display-order relationship relative to the picture(s) used for its prediction.
  - The heightened flexibility of information retrieval means that B-frames typically require fewer bits for encoding than either I or P-frames.

A decoder is mainly comprised of motion compensation, intra frame decoding, inverse DCT, de-quantization, entropy decoder, and de-blocking filter. Once receiver gets a data at the receiver end by users, it is inputted to the entropy decoder. The entropy decoder converts the code words into its respective symbols. Those symbols are converted in the serial stream of 1-by-(m*n), The serial stream is converted into a matrix, which is known as the received quantized-DCT matrix of m-by-n form. The output of entropy decoder is applied to a de-quantization block of decoder. The de-quantization is done with multiplication of received quantized-DCT matrix and quantized co-efficient matrix.

The resultant matrix of de-quantized block is applied to the inverse-DCT block. The reconstructed matrix from an inverse-DCT and de-quantization should be same based on theoretical condition, but it will be different. It means that the reconstructed signal is loses partial information in quantization/de-quantization process. This error will be increased after application of inverse DCT on de-quantized matrix. The inverse DCT reconstructs the original residual signal from its DCT coefficient. The output from IDCT is the residual frame.

The H.264 is the block-based video codec standard, so it reduces the reconstructed image quality compare with the original image. The reconstructed image filters out the pixels value of an image to improve the image quality. The de-blocking filter is implemented to remove blocking artifact in this project.

First, the H.264 decoder generates a compensated frame with help of the motion vectors and the reference frame. If the received frame is a P-frame or B-frame then the current frame is reconstructed by adding the compensated frame into the residual frame. If the received frame is the reference frame, an I-frame is reconstructed using intra mode predictions. The reconstructed image is filtered to improve visual quality, and to remove blocking artifact.

Motion Estimation:
Motion estimation block is also known as inter frame prediction. It is a more effective algorithm for compression, because it removes temporal redundancy from the image. The current frame is predicted using previous frames and future frames. The current frame is predicted on macro-block level. Each of the macro-blocks in the current frame are compared with a predefined search window of the past frames or future frames. The coordinate of the matched macro block is saved in the motion vector (MV). The procedure continues with the next macro-block in the current frame and It ends at boundary of the current frame. Figure shows macro-block of current frame (MB in light blue shade) is searched in reference frame (I in yellow shade), within predefined search area (P in gray shade).

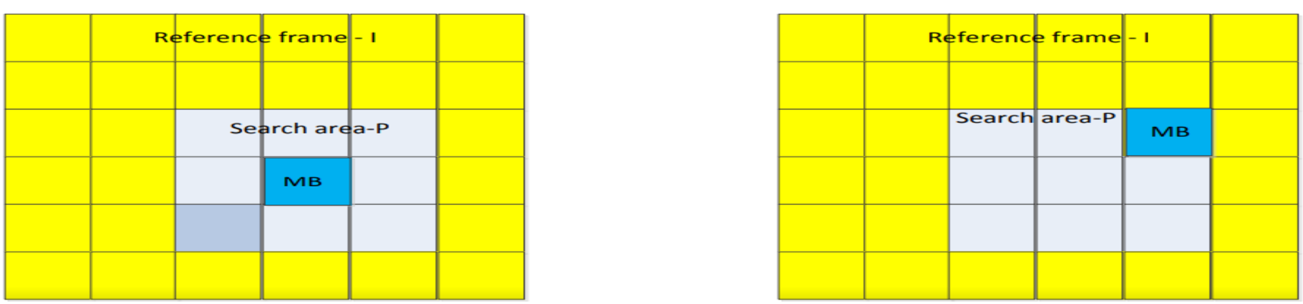

Figure 2: Reference frame (I), search area (P), current macro-block (MB)

Motion Compensation: Motion compensation represents a motion in terms of renovation of a reference frame into a current frame. It means that the motion compensation block takes input as a motion vector and reference frame, and gives the output as a compensated frame. The reference frame might be a previously coded frame or a future frame. The difference between the current frame and the compensated frame is known as the residual frame. The compensated frame is not an exact match with the current frame. The residual image still contains a small amount of information, which is needed at the decoder side to reconstruct a current frame.

Inter Frame Prediction:
Inter frame prediction is used to achieve a higher compression ratio in video codec standards. It takes advantage of temporal redundancy between multiple frames. The current frame is predicted from multiple reference frames. The frames are divided into a number of macro-blocks (MB) before processing. The inter frame prediction module searches the position of the current macro block of the current frame in the reference frame. It uses a matching algorithm to find the maximum match. The motion search is trying to locate the maximum match of the current MB value in the reference frame window, and stores the x-y positions of the current MB. The x-y coordinates are also known as the motion vector (MV). This motion vector is passed to the next block for further processing.

Intra Frame Prediction:
Intra prediction is applied to remove spatial redundancy within the frame. Intra mode prediction is selected if there is not enough temporal redundancy in the two frames. So, the current frame is coded by itself. Intra frame prediction is one form of compression that looks at information of an individual frame, and tries to reduce the amount of information with minimum loss and high quality. Intra frame prediction has two inputs called the reconstructed frame and the original frame. The reconstructed intra image is unfiltered and fed back as input to the intra prediction block.

The predicted macro-block is generated with respect to the reconstructed macro block and the intra mode. The predicted macro-block is subtracted from the original macro-block, and resulted as the residual macro block. The residual macro block is further processed by discrete cosine transform and quantization. The output of quantization is applied to inverse quantization, and inverse DCT, and then generates the residual macro-block. The reconstructed residual macro-block is used as reference for next macro-block in queue.

Deblocking Filter:
When H.264 decompresses the received image, the blocking artifact can be observed in the reconstructed image. Due to block based DCT and quantization, the frequency components are changed abruptly near the block boundaries. The block boundaries can be identified by HVS in the uncompressed image and video. It is known as a blocking artifact. Filter is used to enhance the image quality, remove blur, and increase PSNR. The popular de-blocking filter is implemented in H.264 video codec standard.

IV. IMPLEMENTATION DETAILS

Performance analysis is done by calculating the PSNR value. Peak Signal to Noise Ratio PSNR is an approximation to human perception of reconstruction quality. It is calculated based on the Mean Square Error (MSE). PSNR value is high for enhanced Image. PSNR is a commonly used measure of image fidelity the degree to which one image matches another, such as in comparing a processed image to an original (non-compressed) image.

The phrase Mean Square Error, often abbreviated MSE (also called PSNR, Peak Signal to Noise Ratio) is an engineering term for the ratio between the maximum possible power of a signal and the power of corrupting noise that affects the fidelity of its representation.

$$MSE = \frac{1}{mn}\sum_{i=1}^{m}\sum_{j=1}^{n}\left[I(i,j) - K(i,j)\right]^2$$

the PSNR is defined as

$$PSNR = 20 \times \log_{10}\left(\frac{Max_I}{\sqrt{MSE}}\right)$$

The compression ratio is defined as follows:

$$Cr = \frac{n1}{n2},$$

where n1 is the data rate of original image and n2 is that of the encoded bit-stream.

Video Quality Metric (VQM):
VQM is developed by ITS to provide an objective measurement for perceived video quality. It measures the perceptual effects of video impairments including blurring, jerky/unnatural motion, global noise, block distortion and color distortion, and combines them into a single metric. The testing results show VQM has a high correlation with subjective video quality assessment and has been adopted by ANSI as an objective video quality standard. VQM takes the original video and the processed video as input and is computed as follows:
• Calibration: This step calibrates the sampled video in preparation for feature extraction. It estimates and corrects the spatial and temporal shift as well as the contrast and brightness offset of the processed video sequence with respect to the original video sequence.
• Quality Features Extraction: This step extracts a set of quality features that characterizes perceptual changes in the spatial, temporal, and chrominance properties from spatial temporal sub-regions of video streams using a mathematical function.
• Quality Parameters Calculation: This step computes a set of quality parameters that describe perceptual changes in video quality by

comparing features extracted from the processed video with those extracted from the original video.

• VQM Calculation VQM is computed using a linear combination of parameters calculated from previous steps. VQM can be computed using various models based on certain optimization criteria.

V. RESULTS AND ANALYSIS

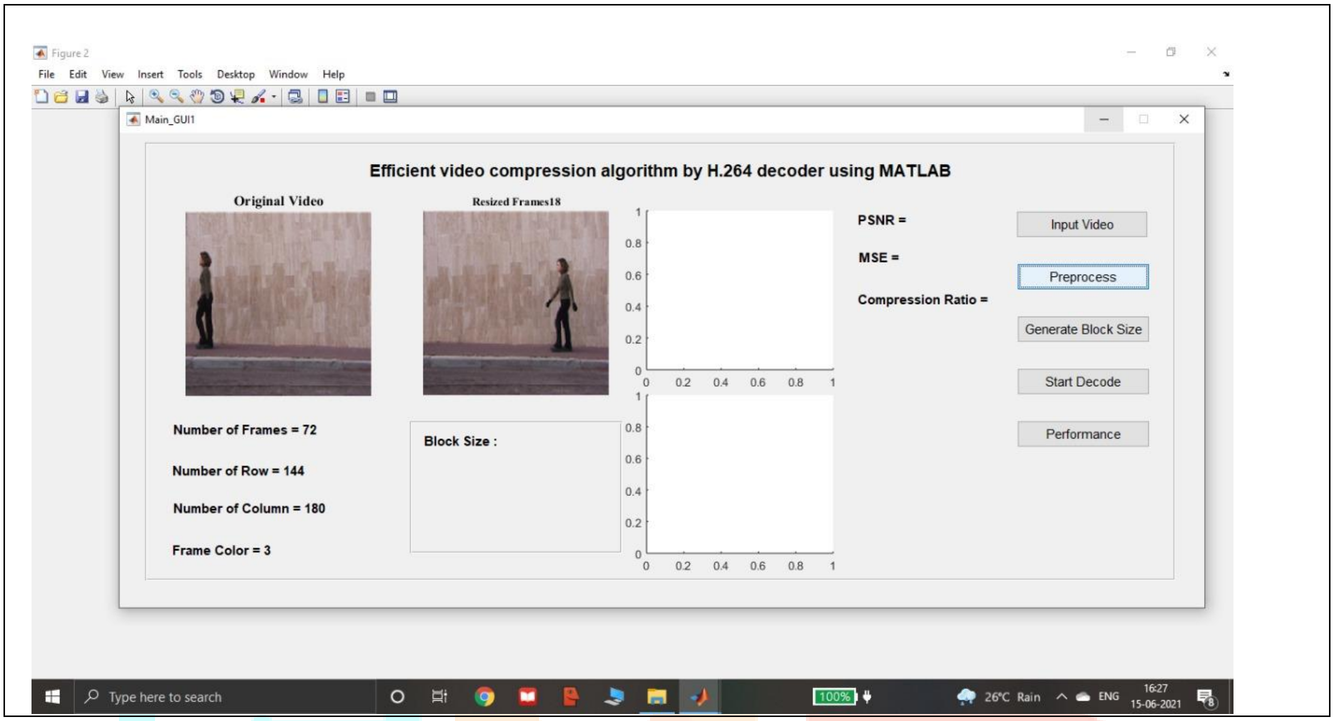

Figure 3: Input video applied to the decoder

The figure 3 shows by feeding the input video, a dialog box is opened to browse the video and select the video. It analyses and provides number of frames, resolution of video (i.e., rows and columns) and frame color on GUI interface. Figure 4 shows the segmented video. In this step the video is divided into frames and those frames are resized accordingly

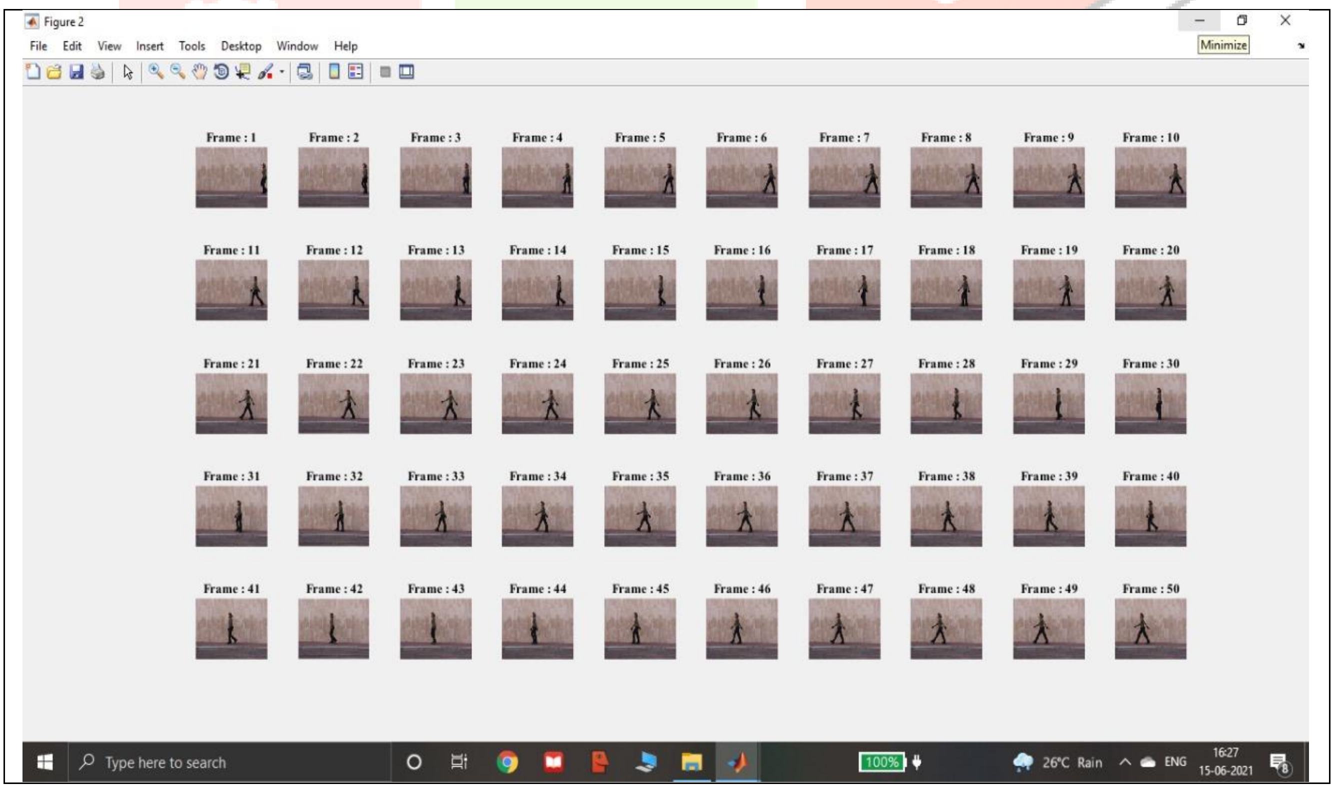

Figure 4: Segmented Video

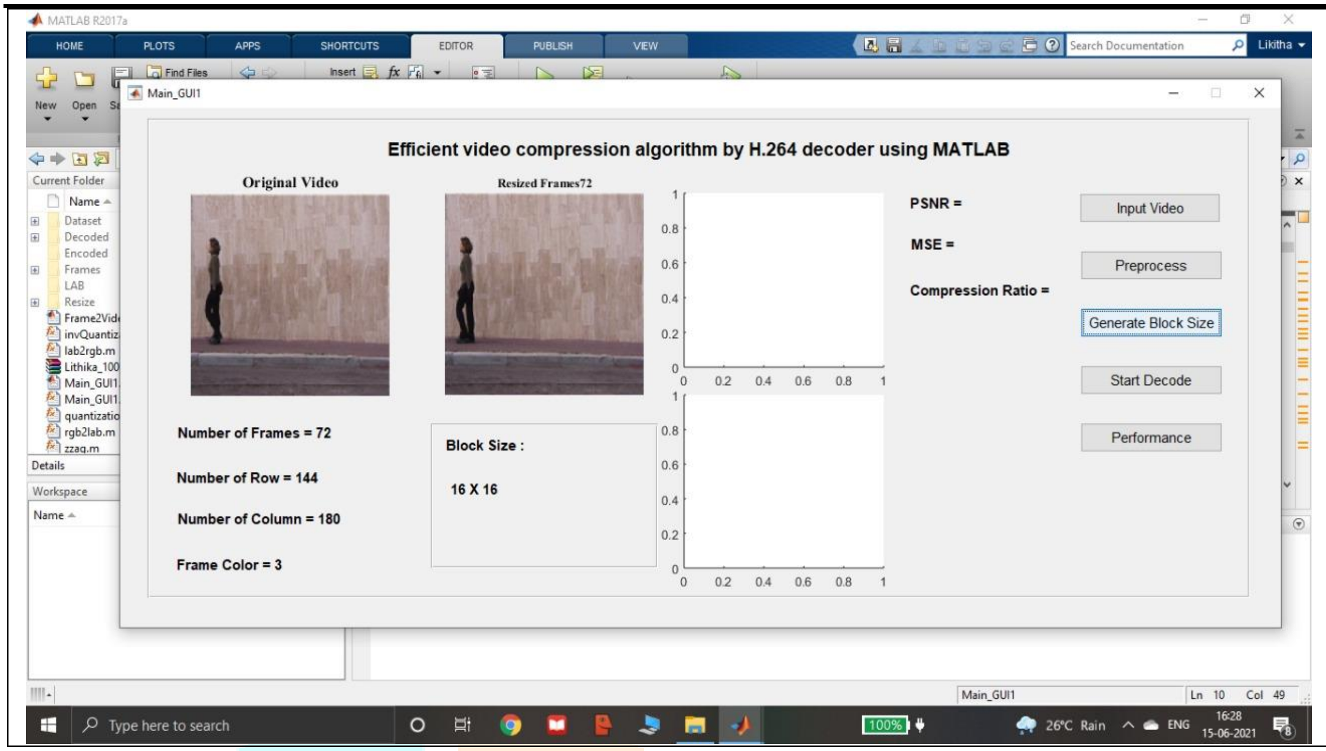

Figure 5: Screen shot of Generating the block size

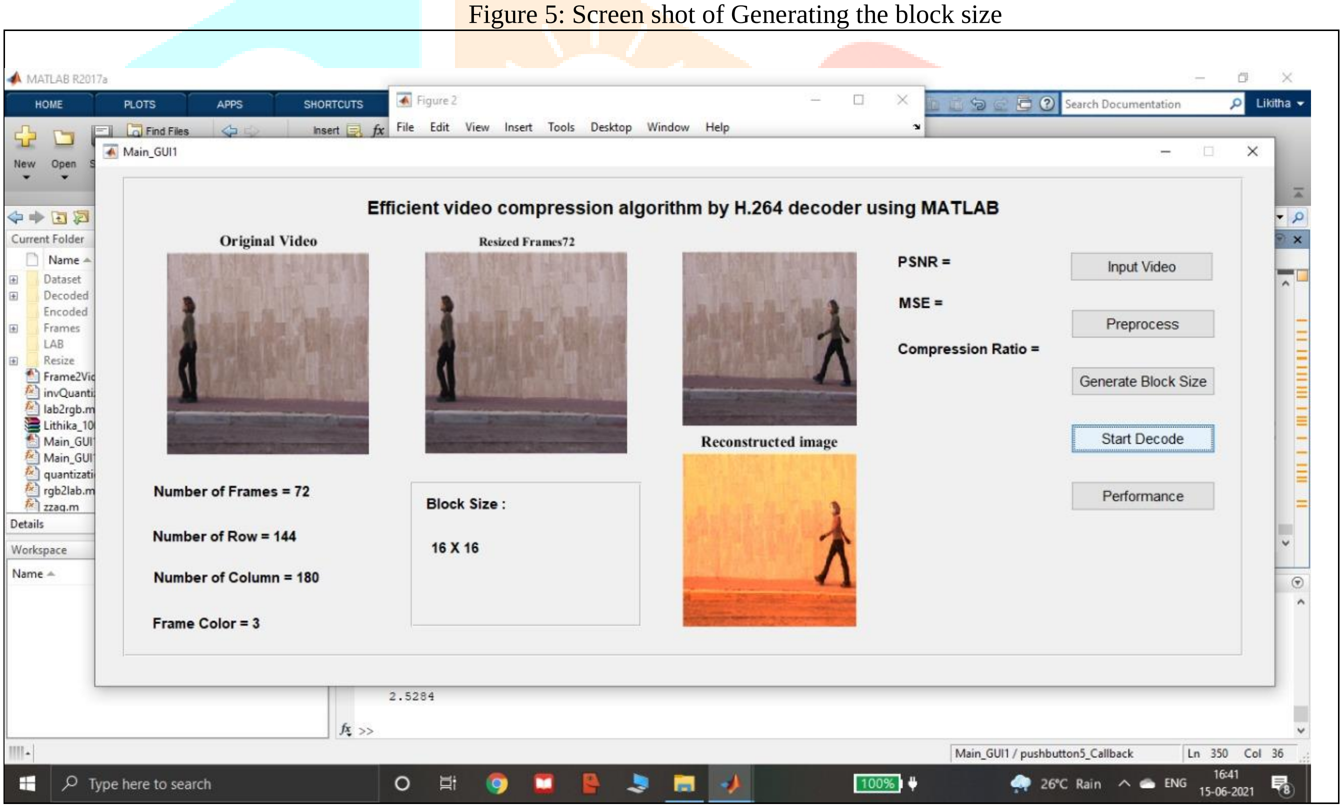

Figure 6: Screenshot of the segmented video

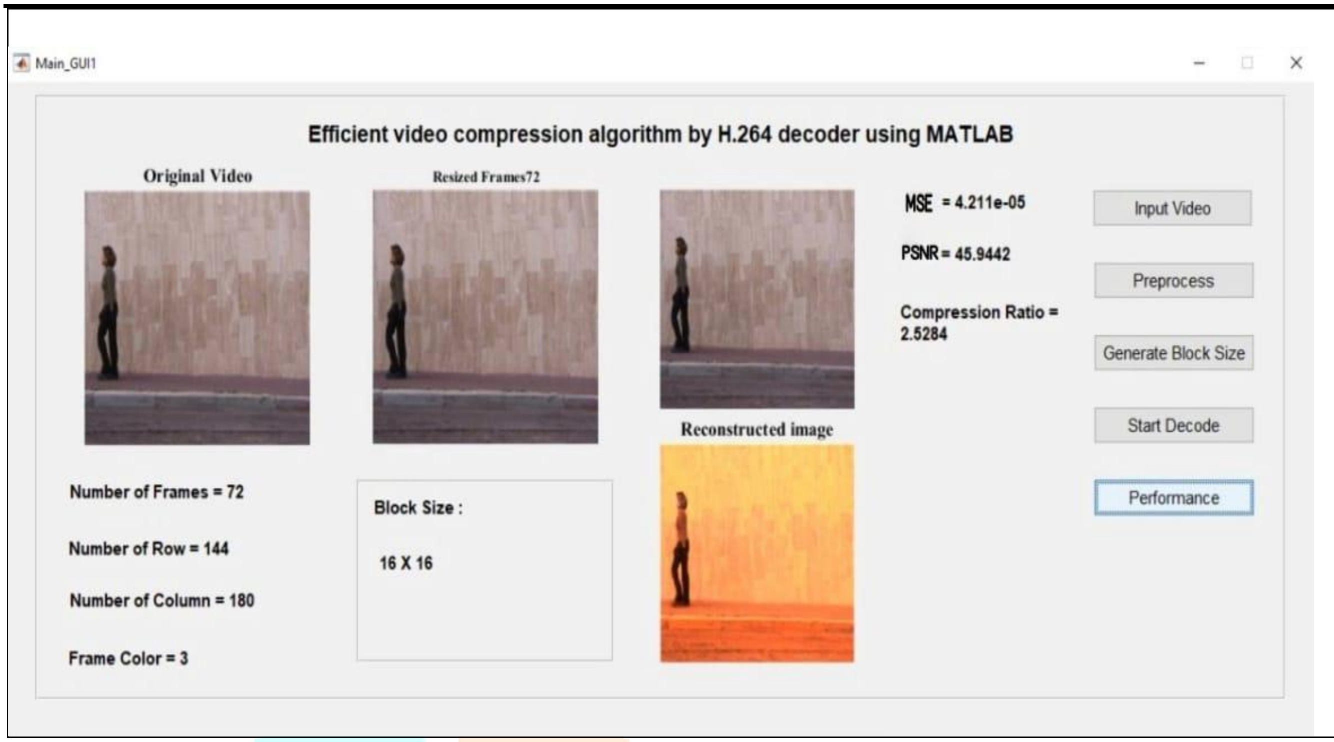

Figure 7: MSE, PSNR and Compression Ratio of the output video

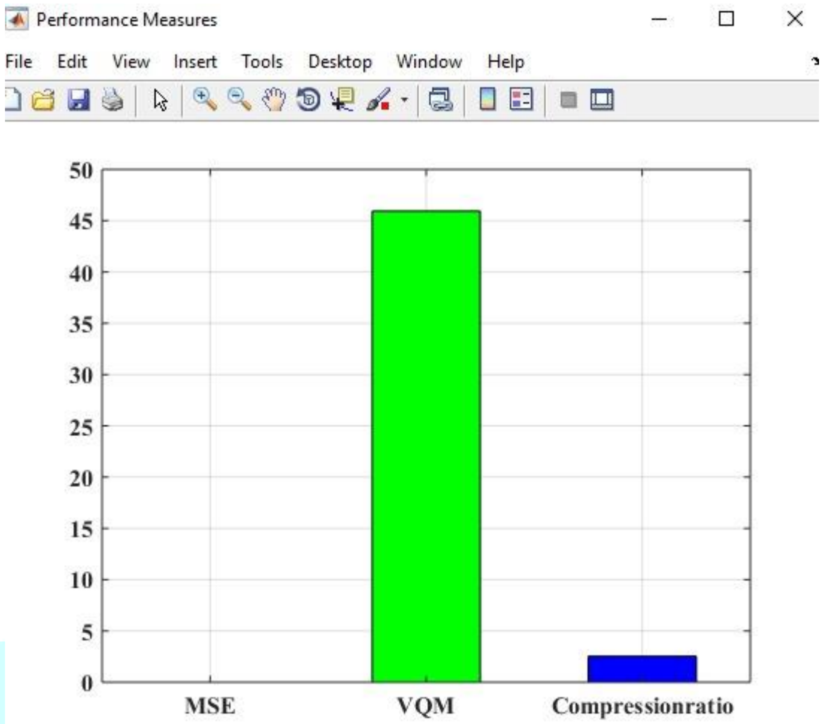

Figure 8: Plot of MSE, PSNR and VQM

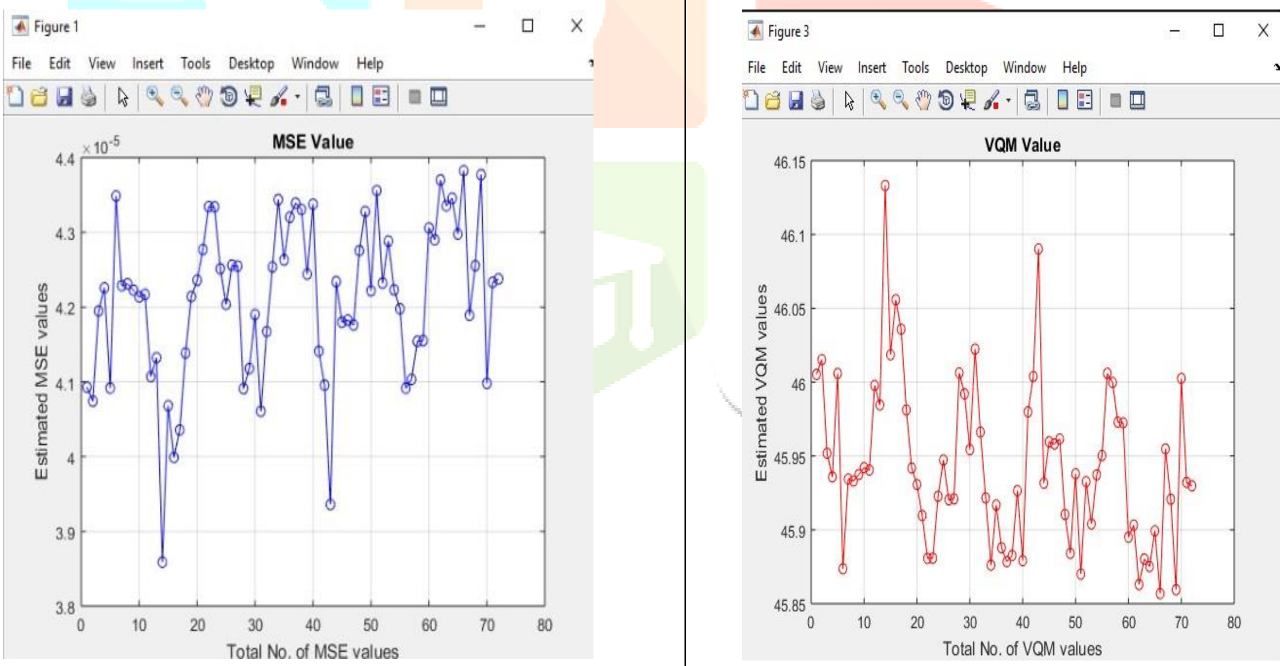

Figure 9: MSE and PSNR for all the frames

V. CONCLUSIONS

The proposed work shows the decoder simulation and step by step analysis of each blocks of the decoder. Also the measurement of the performance of the decoder is measured in-terms of MSE, PSNR and Compression Ratio.

VI. REFERENCES

1. T. Wiegand, G. J. Sullivan, G. Bjøntegaard, and A. Luthra, ‘‘Overview of the H.264/AVC video coding standard,’’ IEEE Trans. Circuits Syst. Video Technol., vol. 13, no. 7, pp. 560–576, Jul. 2003.
2. G. J. Sullivan, J.-R. Ohm, W.-J. Han, and T. Wiegand, ‘‘Overview of the high efficiency video coding (HEVC) standard,’’ IEEE Trans. Circuits Syst. Video Technol., vol. 22, no. 12, pp. 1649–1668, Dec. 2012.
3. Z. Zhao and P. Liang, ‘‘A statistical analysis of H.264/AVC FME mode reduction,’’ IEEE Trans. Circuits Syst. Video Technol., vol. 21, no. 1, pp. 53–61, Jan. 2011.
4. X. Lu and G. R. Martin, ‘‘Fast H.264/SVC inter-frame and inter-layer mode decisions based on motion activity,’’ Electron. Lett., vol. 48, no. 2, pp. 84–86, 2012.
5. Y.-H. Sung and J.-C. Wang, ‘‘Fast mode decision for H.264/AVC based on rate-distortion clustering,’’ IEEE Trans. Multimedia, vol. 14, no. 3, pp. 693–702, Jun. 2012.
6. H. H. Tong and A. N. Venetsanopoulos, “A perceptual model for JPEG applications based on block classification, texture masking, and luminance masking,” in Proc. Int. Conf. Image Process. (ICIP), 1998, pp. 428–432.
7. G. Yammine, E. Wige, and A. Kaup, “A no-reference blocking artifacts visibility estimator in images,” in Proc. Int. Conf. Image Process. (ICIP), Sep. 2010, pp. 2497–2500.
8. K.-H. Thung, R. Paramesran, and C.-L. Lim, “Content-based image quality metric using similarity measure of moment vectors,” Pattern Recognit., vol. 45, no. 6, pp. 2193–2204, 2012.
9. L. Li, H. Zhu, G. Yang, and J. Qian, “Referenceless measure of blocking artifacts by Tchebichef kernel analysis,” IEEE Signal Process. Lett., vol. 21, no. 1, pp. 122–125, Jan. 2014.
10. E. Nadernejad, N. Burini, and S. Forchhammer, “Adaptive deblocking and deringing of H.264/AVC video sequences,” in Proc. IEEE Int. Conf. Acoust., Speech, Signal Process. (ICASSP), May 2013, pp. 2508–2512
11. L. Li, H. Zhu, G. Yang, and J. Qian, “Referenceless measure of blocking artifacts by Tchebichef kernel analysis,” IEEE Signal Process. Lett., vol. 21, no. 1, pp. 122–125, Jan. 2014.
12. G. T. Narayanan, “A study of probability distributions of DCT coefficients in JPEG compression,” Ph.D. dissertation, Dept. Elect. Comput. Eng., New Jersey Inst. Technol., Newark, NJ, USA, 2010.
13. Dr. H S Prasantha, “NOVEL APPROACH FOR IMAGE COMPRESSION USING MODIFIED SVD”, International Journal of Creative Research Thoughts (IJCRT), Volume 8, Issue 8, Page 2234-2243, Aug 2020
14. Dr. H S Prasantha, “IMPLEMENTATION OF IMAGE COMPRESSION USING FAST COMPUTATION OF SVD ON DM642”, International Journal of Creative Research Thoughts (IJCRT), Volume 8, Issue 8, Page 2364-2368, Aug 2020
15. Prasantha, H, H Shashidhara, K N B Murthy, and M Venkatesh. "Performance Evaluation of H.264 Decoder on Different Processors." International Journal on Computer Science & Engineering. 1.5 (2010): 1768. Web. 7 Apr. 2013.
16. Chirag B., A. Lohith and Prashantha H. S. , “Comparative performance analysis of various digital modulation schemes in AWGN channel,” in Innovations in Power and Advanced Computing Technologies (i-PACT), Vellore, India, 2017, pp. 1-5
17. M. J. Raghavendra, H. S. Prasantha and S. Sandya, "Image Compression Using Hybrid Combinations of DCT SVD and RLE", International Journal of Computer Techniques, vol. 2, no. 5, 2015.
18. M.J. Raghavendra, H. S. Prasantha and S. Sandya, "DCT SVD Based Hybrid Transform Coding for Image Compression", International Journal of Recent and Innovative Trends in computing and communication, 2015.
19. H. S. Prasantha, H. L. Shashidhara, and K. N. Balasubramanya Murthy. Image compression using SVD. In Proceedings of the International Conference on Computational Intelligence and Multimedia Applications, pages 143–145. IEEE Computer Society, 2007.
20. Gunasheela K S, H S Prasantha, “Compressive sensing for image compression: survey of algorithms”, Proceedings of Emerging Research in Computing, Information, Communication and Applications, ERCICA, Springer publication, Bengaluru, 2018
21. K N Shruthi, B M Shashank, Y. SaiKrishna Saketh, H.S Prasantha and S. Sandya, "Comparison Analysis Of A Biomedical Image For Compression Using Various Transform Coding Techniques", IEEE, pp. 297-303, 2016