A DEEP-LEARNING-BASED SECURE VIDEO COMPRESSIVE SENSING SCHEME

Thesis submitted
In partial fulfillment of requirements
for the award of a degree of

MASTER OF ENGINEERING
IN
ELECTRONICS AND TELECOMMUNICATION ENGINEERING

By

MOINAK MONDAL

Registration no: 154091 of 2020-2021
Examination Roll no: M4ETC22022

Under guidance of

Dr. Ananda Shankar Chowdhury

Department of Electronics and Telecommunication Engineering
Jadavpur University
Kolkata – 700032
West Bengal
June, 2022

DECLARATION OF ORIGINALITY AND COMPLIANCE OF ACADEMIC ETHICS

I hereby declare that this thesis contains a literature survey and original work by the undersigned candidate, as part of his Master of Engineering in Electronics and Telecommunication in the Department of Electronics and Telecommunication Engineering. All information in this document has been obtained and presented by academic rules and ethical conduct. I also declare that I have thoroughly cited and referenced all material and findings which are not original to this research, as provided by these rules and conduct.

Name: Moinak Mondal

Examination Roll no. 154091 of 2020-21

Registration no. M4ETC22022

Signature of the candidate

FACULTY OF ENGINEERING & TECHNOLOGY JADAVPUR UNIVERSITY

CERTIFICATE OF RECOMANDATION

This is to certify that the thesis entitled — "**A DEEP LEARNING-BASED SECURE VIDEO COMPRESSIVE SENSING SCHEME**" has been carried out by **Moinak Mondal** bearing Class Roll No: **002010701022**, Examination Roll No.: **M4ETC22022** and Registration No: **154091 of 2020-21,** under my guidance and supervision and be accepted in partial fulfillment of the requirement for the degree of Master of Engineering in Electronics and Telecommunication in the Department of Electronics and Telecommunication Engineering.

Prof . Ananda Sankar Chowdhury
Supervisor
Department of Electronics and Telecommunication Engineering
Jadavpur University
Kolkata – 700032

Prof. Ananda Sankar Chowdhury
Head of the Department,
Department of Electronics and Telecommunication Engineering
Jadavpur University, Kol-700032

Prof. Chandan Mazumdar
Dean,
Faculty Council of Engineering
Jadavpur University
Kolkata -700032

FACULTY OF ENGINEERING & TECHNOLOGY

JADAVPUR UNIVERSITY

CERTIFICATE OF APPROVAL

The forgoing thesis titled − "**A DEEP LEARNING-BASED SECURE VIDEO COMPRESSIVE SENSING SCHEME"** is here approved as a creditworthy study of an engineering subject conducted and presented satisfactorily to warrant its acceptance as a precondition to the degree for which it was submitted. It is understood that the undersigned does not automatically support or accept any argument made, opinion expressed, or inference is drawn in it by this approval, but only approves the thesis for the reason for which it was submitted.

Committee on Final Examination for

Evaluation of the Thesis

Signature of External Examiner

Signature of Supervisor

ACKNOWLEDGEMENT

This thesis entitled "**A DEEP LEARNING-BASED SECURE VIDEO COMPRESSIVE SENSING SCHEME**" is the result of the work whereby I have been accompanied and supported by many people, my guide, my friends, and lab seniors. It is a pleasant aspect that now I have the opportunity to express my gratitude to all of them.

First and Foremost, with great respect and appreciation, I would like to extend my gratitude to my PG project guide, **Prof. (Dr.) Ananda Sankar Chowdhury** for his continuous assistance and supervision during the entire duration of this project. His advice and suggestions were instrumental in generating the concept for this project.

Secondly, I would like to express my thanks to **Mr. Jagannath Sethi**, Ph.D. candidate, for his relentless co-operation and constant monitoring throughout the entire session.

Thirdly, I'd want to thank all the members of the Image, Vision & Pattern Recognition (IVPR) group for their invaluable advice and constant support during the development of this project.

Lastly, we would thank my alma mater, Jadavpur University, for encouraging me to pursue a Project involving large-scale research and associated studies.

Signature of the candidate

ABSTRACT

With the advancement of technology and the increased use of video-based communication, we must deal with both massive volumes of information and highly sensitive data in the form of video. As a result, video data must be stored, accessed, and processed in a safe, efficient, and effective way. For that purpose, we first introduce the notion of compressive sensing in this study and then offer a deep-network-based compressed sensing approach that investigates both temporal and spatial correlation of video during signal restoration by employing compensation through multilayer deep features. We also offer a unique encryption approach for protected transmission of sampled video frames based on chaotic sequence and maximum distance separable (MDS) matrices.

Keywords: Compressive sensing, Convolutional neural network, Multilevel feature, Encryption, Maximum distance separable (MDS) matrices.

TABLE OF CONTENTS

LIST OF FIGURES

Figure no.	List of figures	Page no.

LIST OF TABLES

CHAPTER 1: INTRODUCTION

The fast expansion of digital multimedia transmission through wireless networks and the internet has resulted in a rise in multimedia data consumption. Simultaneously, as the resolution of cameras has risen, the bulk of video data has grown, posing a problem for today's network in terms of data processing capacity and bandwidth. In addition, standard video codecs such as HEVC, h.264, and others need high computing power for video compression. Because significant computational capacity is not available in wireless sensor networks, compressive sensing-based video compression is a viable option for compressing video data because it is less difficult than classical video compression used in above-mentioned codecs.

Compressive Sensing [1] is a novel approach for compressed signal sampling and reconstruction in which the sampling is accomplished using matrix multiplication. The matrix is referred to as a measurement matrix. To reconstruct such a signal an optimization problem is performed, using auxiliary conditions specified by an underdetermined system of linear equations. In signal reconstruction, the measurement matrix is very important. So, if a secure measurement matrix meets certain criteria, it may be used as a key to safeguard data from illegal access or alteration.

This thesis starts with an overview of compressive sensing theory before moving on to detail the many applications of this theory in the field of video signal processing. Later on, it shows how deep learning has been used to showcase the present use of compressive sensing in video compression, and how it has lowered reconstruction time while also improving video quality for a higher compression ratio. It then goes on to discuss our proposed network for compressed video signal sensing, as well as

a new compression-independent encryption mechanism for safe video signal transfer.

The study then goes on to describe the experimental findings gained for several deep network topologies, before concluding with a discussion of how the network might be enhanced further.

CHAPTER 2: LITERATURE REVIEW

2.1 Compressive Sensing Theory

Compressed sensing is a technique used in signal processing for efficiently collecting and recreating a signal by solving underdetermined linear equations [2]. The sparsity of the signal is considered while reconstructing the original signal using a lower number of samples than the number of samples required in the Nyquist-Shannon theorem.

The majority of natural signals, such as pictures and sounds, can be compressed very efficiently. Signal compressibility refers to the ability of a signal to be represented on a basis with just a few active modes, hence lowering the number of active modes necessary to properly represent a signal. A signal $x \in R^n$ is compressible mathematically if it can be represented as a sparse vector $s \in R^n$ (mainly contains Zeros) in a transform basis $\Psi \in R^{n \times n}$

$$x = \Psi \cdot s \tag{2.1}$$

If a vector s has precisely K nonzero items in Ψ domain, then it's K-sparse. If the basis Ψ is general, like Fourier or wavelet, only a few active terms in s are needed to reconstruct the original signal x, minimizing the data needed to store or transmit the signal.

Compressed sensing utilizes a signal's sparsity on a generic basis (Ψ) to reconstruct it from significantly fewer samples than the Nyquist rate. That is if a signal $x \epsilon R^n$ is $K - saprse$ in domain Ψ, then a random measurement $y \epsilon R^p$ of the signal will be able to generate its sparse representation (s) in the transformed coordinate system (Ψ) which will eventually will be able to recover the original signal (x). The measurement can be mathematically represented as $y \epsilon R^p$, where $K < p \ll n$.

$$y = \Phi \cdot x \tag{2.2}$$

Here, $\Phi \epsilon R^{p \times n}$ is measurement matrix which represents a collection of p linear measurements of the state x.

It is feasible to reconstruct the signal x if the sparse vector s is known (2.1). Compressed sensing's purpose is to discover the sparsest vector s that is compatible with the measurement data of y.

$$y = \Phi \cdot \Psi \cdot s \tag{2.3}$$

Because there are infinitely many consistent solutions of s as the system of equations in (1.3) is underdetermined. The sparsest solution for s is found by solving the optimization equation below:

$$\hat{s} = \underbrace{argmin}_{s} \|s\|_0 \; s.t. \; y = \Phi \cdot \Psi \cdot s \tag{2.4}$$

Where $\|\cdot\|_0$ is the l_0 pseudo-norm, which is defined as the number of nonzero entries; this is also known as the cardinality of s.

This NP-hard optimization issue in (2.4) can only be solved using a combinatorial n and K brute-force search. If level of sparsity K is unknown, the search is significantly larger. This combinatorial search makes solving (2.4) intractable for even modestly large n and K which require exponentially rising computing capacity.

Fortunately, under certain situations, (2.4) may be relaxed to a convex $l_1$minimization.

$$\hat{s} = \underbrace{argmin}_{s} \|s\|_1 \; s.t. \; y = \Phi \cdot \Psi \cdot s \tag{2.5}$$

Where $\|\cdot\|_0$ is the l_1 norm, which is represented as,

$$\|s\|_0 = \sum_{k=1}^{n} |s_k| \tag{2.6}$$

For the l_1-minimization in (2.5) to converge with high probability to the sparsest solution in the optimization problem (2.4) some extremely stringent requirements [2,3] must be satisfied as follows,

1. It is essential that the measurement matrix, denoted by Φ, be incoherent with regard to the transform domain Ψ. This indicates that the rows of Φ should not be correlated with the columns of Ψ. Example of such structurally random matrix which follows such properties are Gaussian matrix, Rademacher matrix, Hadamard matrix etc.

2. The number of measurements, denoted by p, ought to be of adequate size, about equivalent to

$$\mathrm{p} \approx \sigma\left(\mathrm{K}\log\left(\frac{\mathrm{n}}{\mathrm{K}}\right)\right) \approx k_1 \cdot \mathrm{K}\log\left(\frac{\mathrm{n}}{\mathrm{K}}\right) \tag{2.7}$$

The constant multiplier k_1 depends on the incoherency between Φ and Ψ .

2.2 Video compressed sensing

2.2.1 Video data Acquisition:

Mainly two techniques are followed in video data accusation by compressed sensing, these are discussed in the following sections.

2.2.1.1. Frame by frame Acquisition:

Each frame of a video is treated as a individual images for this method. Converting a 2D frame into 1D vector allows for a simple acquisition of frame data [4]. The method can be mathemetically shown as below two step

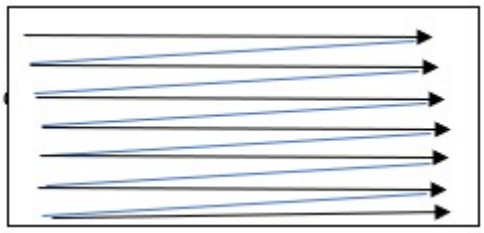

Fig1.1: Raster scanning of frames

(1) In the first step N × N image X is Rasterized into an $N^2 \times 1$ -dimensional vector x:

$$x = Raster(X) \qquad (2.8)$$

Here, $Raster(\cdot)$ is a rasterization method—Which is performed by, the concatenating N rows of the video frame together, later a transpose is performed, which gives a $N^2 \times 1$ vector.

(2) Now for sensing compressively, we apply $M \times N^2$ measurement matrix

$$y = \Phi_k \cdot x \qquad (2.9)$$

From above we get compressed sensed data y which is a one-dimensional column vector having size $M \times 1$. The objective of compressed sensing is to recover x from the measurement y.

2.2.1.2. Block-based video Acquisition:

Another method is to present video frames as a collection of groups of pictures (GOP), with the one frame having more information or correlation with other frames is treated as key frame and the rest of the frames are sent as non-key frames or dependent frame. In the sensing portion of this method the compression ratio (higher Sub rate) is kept lower for keyframe and higher for the non-key frame to achieve significant compression. The method is as follows,

(1) We each video frame of size $I_r \times I_c$ is partitioned into K nonoverlapping blocks of size B × B, here each block is represented as a vectorized column of length $N(= B^2)$.

(2) Then these column vectors obtained in the first step are then compressed individually by projecting onto random measurement matrix Φ_K or $\Phi_{NK}\, for$ non-key frames [5].

$$y_K = \Phi_K \cdot x_{K,\ k}\ \mathrm{k} = 1,2,3 \ldots\ldots \mathrm{K} \tag{2.10}$$

$$y_{NK} = \Phi_{NK} \cdot x_{NK,\ k}\ \mathrm{k} = 1,2,3 \ldots\ldots \mathrm{K} \tag{2.11}$$

where $x_{K,\ k}$ and $x_{NK,k}$ denotes the kth block of key frame and non-key frame, respectively.

The size of Φ_K is $M_k \times$ N such so the key-frame sub rate is denoted as $S_k = M_k/N$, and the size of Φ_{NK} is $M_{nk} \times$ N such that the non-key frame sub rate is given as $S_{nk} = M_{nk}/N$.

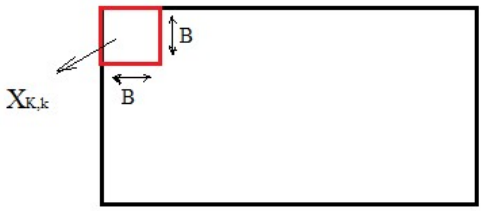

Fig1.2: Block Compressed Sensing

The whole measurement matrix in this situation may be expressed in Block diagonal form as:

$$\Phi = \begin{bmatrix} \Phi_B & 0 & \ldots \\ 0 & \Phi_B & 0 \\ \vdots & \vdots & \ddots \\ 0 & \ldots & \Phi_B \end{bmatrix}$$

The key frame sub rate is kept sufficiently high such that the frame can be recovered from any standard CS recovery algorithm [4,5]. Then the non-key frames are recovered using many prediction methodologies of compressed sensed videos which is discussed later.

2.2.2. CS Video reconstruction:

The goal of compressed sensing video reconstruction is to minimize the error between the original frame(x) to that of the reconstructed frame $(\hat{x})$, this can be mathematically shown as,

$$\hat{\mathrm{x}} = \underbrace{argmin}_{\hat{x}} \ \|x - \hat{x}\|_2 \tag{2.12}$$

But since in the prediction or the decoder side there is no original signal (x). So, to approximate the above minimization problem the known measurement data (y) can be used in two ways in the above-mentioned minimization problem. Those are,

1. The initial recovery of y can be used to approximate the original data (x).

$$\hat{\mathrm{x}} = \underbrace{argmin}_{\hat{x}} \ \|Reconstruction(y, \Phi) - \hat{x}\|_2 \tag{2.13}$$

   Where, $Reconstruction(\cdot)$ is any standard CS recovery method. The obtained (x) in above and using the residual the original data can be recovered. [6]

2. The optimization problem of (2.13) can also be changed from signal domain data (x) to measurement domain data(y).
   Which is,

$$\hat{\mathrm{x}} = \underbrace{argmin}_{\hat{x}} \ \|y - \ \Phi\hat{x}\|_2 \tag{2.14}$$

   In case of block-based reconstruction, the above-mentioned problem looks like,

$$\hat{\mathrm{x}}_{f,i} = \underbrace{argmin}_{\hat{x}} \ \left\|y_{f,i} - \ \Phi_B\hat{\mathrm{x}}_{f,i}\right\|_2 \tag{2.15}$$

Where f is the index of the frame and i is the index of the block and Φ_B is the measurement matrix for that block.

All the reconstruction algorithms try to solve the minimization problems mention in the above section. This section discusses classic CS video reconstruction techniques that are commonly used.

2.2.2.1. Straightforward CS Reconstruction for Video:

This is the simplest method for implementing CS on video, which would be to convert a series of 3D frames into a vector in 1D. Because of this, vectorization of 2D pictures has computational and memory challenges that are magnified when applied to video data. Here, every frame is considered as a separate image, therefore the inter-frame correlation between subsequent video frames is ignored. This leads in inefficient video data gathering. To reduce computation and memory complexity, frame-by-frame block-based image measurement technique is used for video data gathering.

In [5] BCS-SPL algorithm is proposed for reconstruction of individual images. Which is utilized to independently rebuild individual video frames from sampling data without considering spatial correlation. This is the most straight forward reconstruction method for videos sampled by compressive sensing.

In [7, 8] discussed 3D-BCS-SPL algorithm which applied to reconstruct 3D volumetric sampled video data by taking spatial and temporal correlation of consecutive video frames. Here they considered BCS-SPL algorithm in a 3D version, by extending BCS-SPL method for restoration an independent image into three dimensions. Partitioning video signal into smaller 3D cubes is followed for

overcoming computational cost and storage limitation issues (which increases as the dimensionality of the signal increases).

2.2.2.2 BCS-SPL using Motion-Compensation:

In order to create inter-frame predictions for efficient prediction residual coding, object motion knowledge is utilized in video coding. Because of this, motion estimation and compensation are a crucial and commonly used part of traditional video-coding methods.

This ME/MC method was introduced in the decoder (in contrast to the standard MC/ME video compression network system, which employs MC/ME in the encoder or compressor side of the whole network) side of the compressive sensing network in [9].

The suggested approach in [9], which incorporates reconstruction from a residual resulting from motion estimation and compensation from a reference frame, iteratively reconstructs frames of the video sequence and their accompanying motion fields. Experiments show that the proposed method is much better than a simple reconstruction that uses a still-image reconstruction independently for each frame.

2.2.2.3. Multi-Hypothesis prediction of video:

Multi-hypothesis (MH) prediction [10, 11] is a kind of Motion compensation technique often used in conventional video coding in which many, unique predictions are made from different references and then merged to give a composite prediction.

By motion compensation [10], 3D recursive search [12] etc. methods are used to obtain structurally similar patches or blocks (in case of block-based reconstruction) in the reference frames to create Hypothesis set.

A hypothesis set $H_{f,i}$ for can be represented as collection of matching blocks in the reference frame for i^{th} block of f^{th}frame as,

$$H_{f,i} = \{h_1, h_2 .. h_2\} \tag{2.16}$$

Using these hypotheses set the prediction of the original frame level data can be shown as the weighted (w) sum of these hypothesis,

$$\hat{x}_{f,i} = w_{f,i} H_{f,i} \tag{2.17}$$

Thus, the optimization problem of (2.7) changes to,

$$w_{f,i} = \underbrace{argmin}_{w} \left\| y_{f,i} - \Phi_B \, w_{f,i} \, H_{f,i} \right\|_2 \tag{2.18}$$

The problem in (2.18) is in updated in later stages by introducing Tikhonov regularization [10,27], hypothesis set update [13], by reweighting the Tikhonov regularization term [14] to obtain better result.

2.2.2.4. Reweighted residual sparsity:

In [15] reweighted residual sparsity method produces best result for comparatively higher sub-rate $(0.1,\ 0.2)$ for some of the test videos. This method is also the extension of the multi-hypothesis model. Additionally, this method updates the weight of each hypothesis based on the mean square error between the predicted patch $(\hat{x}_{l,k}{}^{prev})$ in previous iteration and the matching block or hypothesis $(h_{l,i})$ in the reference frame.

The MSE is given as,

$$MSE\left(\hat{x}_{l,k}{}^{prev}, h_{l,i}\right) = \frac{1}{s^2} \left\| \hat{x}_{l,k}{}^{prev} - h_{l,i} \right\|_2^2 \tag{2.19}$$

The residual is calculated by subtracting the predicted frame from the reconstructed frame in previous iteration. *Split Bregman Iteration* based solving scheme is

considered for solving a weighted l_1 minimization problem which attempts to minimize the probability of residuals' DCT coefficient, which is updated in each iteration. RRS provides best quality restoration of video among the both classical and deep-learning based methods for comparatively higher sub-rate ($0.1, 0.2$).

2.2.2.3 Other classical methods:

Other classical reconstruction methods have mostly applied iterative approach to reconstruct video frames. Which resulted into time inefficient reconstruction. Some of these methods are Total variation [16], BCS-SPL-DCT [17], BCS-SPL-DWT [17], and SGSR [18]. For both image and video compressive sensing networks, recent deep-learning-based reconstruction is preferable due to computation time.

2.3. Deep learning in CS

Emerging image CS approaches based on deep learning provide high restoration quality with little computing cost. Some of the recent methods are discussed below

Traditional optimization-based image CS approaches often construct the network using a data flow model. For instance, in [19] authors suggested a deep network (ADMM-Net) for improving a Compressive Sensing-based magnetic resonance imaging (MRI) model based on the iterative processes of the ADMM algorithm [20]. Authors in [21] has developed a structured deep learning model (ISTA-Net) motivated by the iterative shrinkage thresholding approach to optimize a generic l_1 norm CS restoration framework. However, by optimizing the network as a whole, the end-to-end approach for restoration has increased the quality of restoration. For example, ReconNet [22], a simple convolutional neural network, emphasizes recovery which is noniterative in nature. Authors of [23] have produced a Laplacian

pyramid reconstructive adversarial network (LAPRAN) that creates several outcomes with various resolutions concurrently. In addition, in our base paper [24] the authors have introduced a scalable Convolutional Neural Network for Compressed Sensed images that enables both coarse- and fine-grained scalable sampling and reconstruction with a single model. These papers are mostly used for image restoration. For video data, these may be applied independently to each individual frame without considering inter-frame correlation. Shi et al. [25] present a unique video compressive sensing (VCSNet) using a convolutional neural network that takes both intraframe and interframe correlations into consideration while restoring a video. VCSNet splits the video stream into numerous groups of pictures (GOPs), where the very first frame is considered as key frame captured at a greater sampling ratio in comparison with rest of the video frames of that GOP. This technique outperforms current video Compressed Sensing methods and image Compressive Sensing methods which are based on deep learning in terms of actual and perceived reconstruction performance.

CHAPTER 3: PROPOSED METHOD

In this part, we will first describe the fundamental foundation for video compressed sensing using convolution blocks in deep learning. The next subsections present the specifics of Our proposed video compressed sensing network with only one keyframe and deep feature compensation from the previous non-key frame. Typically, keyframe measurements include greater information than non-key frame measurements. Our network separates the video stream into numerous GOPs, the first of which is a keyframe sensed at a lower compression ratio or higher sub-rate than the remaining non-key frames. Our network gathers measurements by utilizing a convolution layer having unique kernel dimension and stride to perform block-based framewise compressed sampling, which not only decreases computation cost and memory complexity but also allows keyframes and non-key frames to be sensed with various sampling rates.

Then, depending on the size of the compressed data, we offer a new classical encryption module based on chaotic sequence and diffusion. We will discuss our encryption module in the later stages. So, form the output of this stage we get compressed and encrypted data. The basic transmitter side of our proposed scheme loos like in shown in *Fig3.1*.

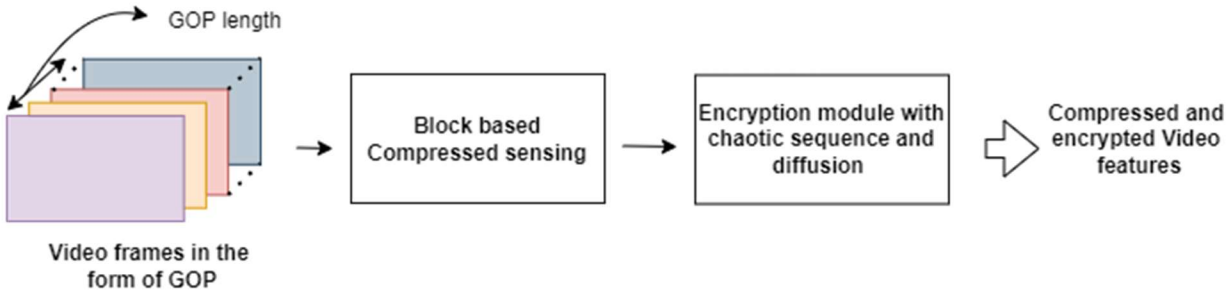

Fig3.1: Basic transmitter side of our network

Then after decrypting the encoded data the reconstruction network is followed. Our network's video reconstruction includes framewise preliminary reconstruction and multilayer feature compensation-based deep reconstruction. Preliminary reconstruction transforms compressed measurement data to frame level data. In deep reconstruction part of the network the deep feature compensation recovers the frame quality by utilizing the key frame deep features along with the deep features of previous non-key frame. Basic block diagram for the reconstruction of the frame data is presented pictorially in *Fig3.2.*

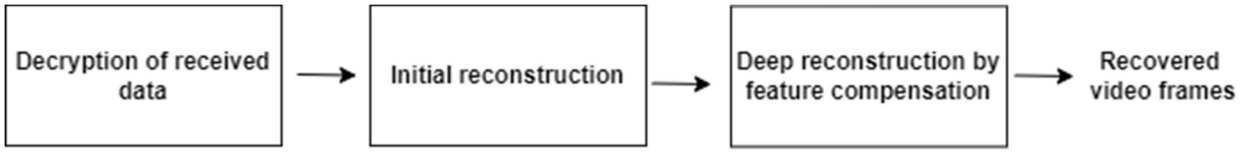

Fig3.2: Basic decoder side of our reconstruction network

Because our deep network is trained simultaneously for both sampling and deep reconstruction not considering the encryption block, the whole deep network for our proposed technique is shown in *Fig3.3.*

Fig3.3: Block diagram of the proposed deep network for video compressive sensing scheme

In our scheme the middle frame of a GOP is used as key frame and other rest of the frames as non-key frame. the idea is to reduce average distance among the key frame and other non-key frames since we know that the correlation reduces between two frames if the distance between increases. The overall descriptive block diagram of our scheme is given in *Fig3.4.*

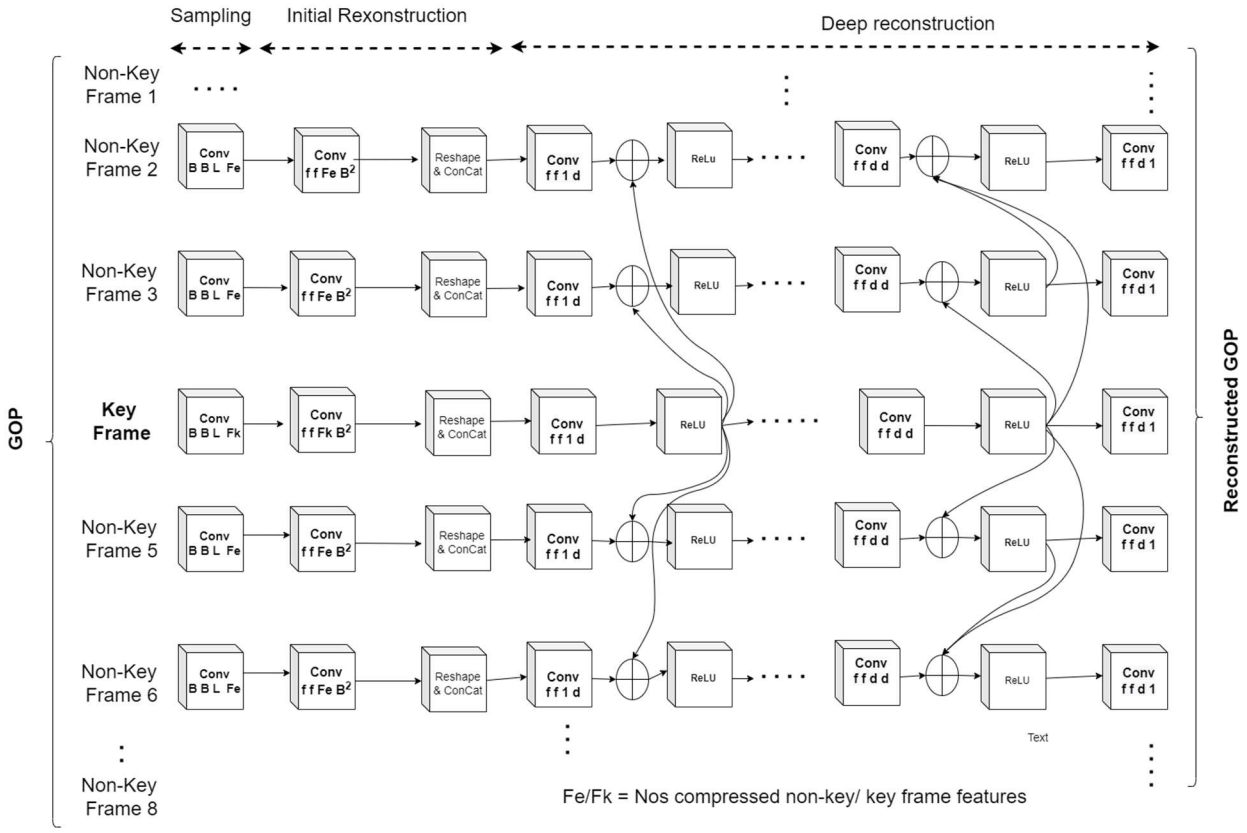

Fig3.4: Descriptive Block diagram of the proposed deep network for video compressive sensing scheme

3.1. Block-based Framewise Sampling:

For hardware usage block sampling is favored over sampling entire sets of spatial and temporal information at once. The *Fig3.5* depicts how a convolution layer is used to sampled the frames in block-based manner.

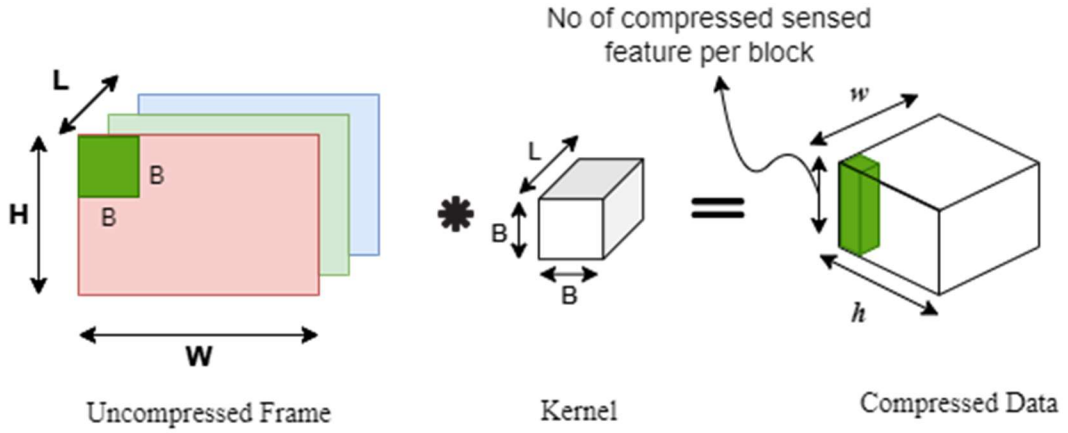

Fig3.5: Block Compressive Sampling process of proposed scheme.

For a sampling ratio α, and a frame having l channel the sampling block size is $B \times B \times L$ and the measurement data of the compressed sensing is given by,

$$y_j = \Phi_B x_j \tag{3.1}$$

Where $\Phi_B \in R^{\alpha l B^2 \times l B^2}$ is the measurement matrix, $x_j \in R^{l B^2 \times 1}$ is the original patch and $y_j \in R^{\alpha l \ ^2 \times 1}$ is the measurement data of j^{th} patch.

Now Φ_B can be replaced with convolutional kernel with special size and stride for key and non-key frames. An image is particularly segmented into $h \times w$ blocks having size $B \times B \times L$ each, and every row of measurement matrix is replaced as a $B \times B \times L$ kernel of the convolution filter. One compressed measurement is obtained by convolving one $B \times B \times L$ kernel across one block. So, to get $\alpha l B^2$ compressed feature from one block we need $\alpha l B^2$ numbers of kernel. The whole frame is convolved using non-overlapping patches of by taking stride of $B \times B$.

So, the sensing of a whole frame by convolution can be mathematically presented for key frame as, $Y_{key} = W_k^s * X_{key}$ (3.2)

for non-key frame as, $Y_{nk,i} = W_{nk,i}^s * X_{nk,i}$ (3.3)

Where Y is the measurement data, X frame level data and W is the weights of convolutional network. The difference between W_k^s and W_{nk}^s is that the no of filters depends on the sub-rate of compression. So, W_k^s has $\alpha_{key} lB^2$ filters and W_k^s has $\alpha_{key} lB^2$ numbers of filters.

3.2. Encryption module:

In this stage we discuss the encryption module and the steps of encryption in detail manner. The block diagram of our proposed encryption module is as follows in *Fig3.6*.

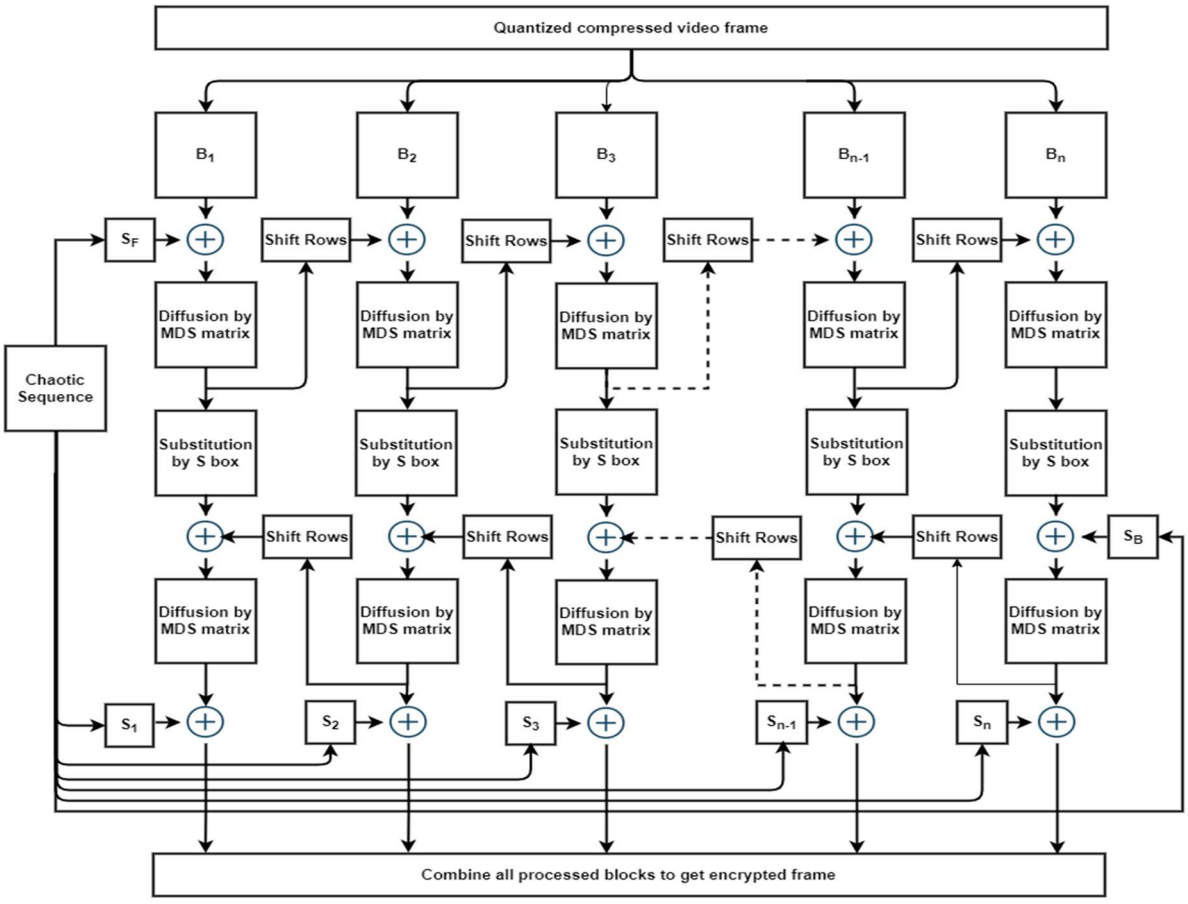

Fig3.6: Detailed block diagram of the Encryption module

Before performing encryption all the compressed feature frames are quantized. In the later stages' encryption of keyframe and non-key frames are performed separately for each GOP.

Steps for achieving encryption are describe below,

1. Divide quantized compressed frames blocks of size 4x4
2. Generate a chaotic sequence and divide it into 4x4 block sizes (there are two more numbers of blocks for the chaotic sequence than that of quantized frames)
3. Perform xor between the current block and its shift rows version of the previous diffused block (left to right) except first block
4. Perform forward diffusion by multiplying AES maximum distance separable (MDS) matrix with the currently processed block
5. Perform substitution using AES substitution box
6. Perform xor between the current block and its shift rows version of the previous diffused block (right to left) except the last block
7. Perform backward diffusion by multiplying AES maximum distance separable (MDS) matrix with the currently processed block
8. Perform xor operation between the processed block and the chaotic sequence block
9. Combine all the processed blocks to get the encrypted frame

This way encryption of compressed data can be achieved by classical method.

For decryption od the encrypted data we perform an inverse producer of XOR between chaotic sequence block and encrypted block. The encrypted video may then be decrypted using inverse backward diffusion and inverse forward diffusion operations.

3.3. Framewise initial reconstruction:

Since, optimizing the images by block reconstruction may give rise to blocking artifacts it is preferable to optimize the images as a whole frame instead of blocks.

Thus, this stage deals with decompression of compressed measurement blocks followed by reshape and concatenation of the blocks to covert the measurement data into frame level. The process is graphically shown in *Fig3.7.*

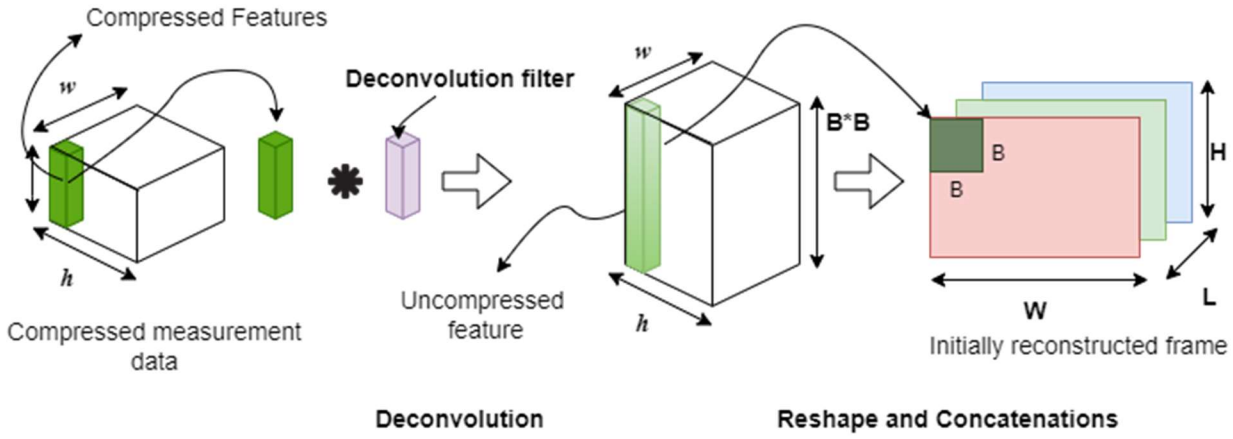

Fig3.7: Detailed flow of the initial reconstruction process

So, the measurement data for each block y_j of size $1 \times 1 \times \alpha LB^2$ size is decompressed or de convoluted into size $1 \times 1 \times LB^2$, which requires LB^2 number of filters of kernel size $1 \times 1 \times \alpha LB^2$. This process can be written mathematically by,

For key frame, $$\check{I}(X_{key,ir}) = W_k^{ir} * Y_{key} \quad (3.4)$$

For non-key frames, $$\check{I}(X_{nk,ir,i}) = W_{nk,i}^{ir} * Y_{key,i} \quad (3.5)$$

Where $\check{I}(\cdot)$ denotes initial restoration of the measurement data, which is pictorially shown in *Fig.* W^{ir} denotes weights of preliminary restoration stage. W_k^{ir} and $W_{nk,i}^{ir}$

has different support length $(1 \times 1 \times \alpha LB^2)$ based on their sub-rates (α_k and α_{nk} respectively) for key and non-key frames.

These reconstructed are in vector form of size LB^2 so, they are first reshaped into blocks of size $B \times B \times L$ then concatenated according to their indices (which is denoted by h and w) to form the frame level initial restoration which is enhanced in the following stage of deep frame enhancement by deep feature compensation from key frame and previous non-key frames.

The reshape and concatenation of the decompressed block mathematically looks like the following equation,

$$\check{X}_* = \varsigma \begin{pmatrix} \mathrm{r}(\check{I}(X_*)^{11}) & \cdots & \mathrm{r}(\check{I}(X_*)^{1w}) \\ \vdots & \ddots & \vdots \\ \mathrm{r}(\check{I}(X_*)^{h1}) & \dots & \mathrm{r}(\check{I}(X_*)^{hw}) \end{pmatrix} \tag{3.6}$$

Where ς is the concatenation function and r is the reshaping function.

3.4. Deep reconstruction:

In a GOP, keyframe and non-key frame measurements are frequently correlated temporally, and the keyframe measures include more information than the non-key frame data. As a result, keyframe's deep features are utilized to correct non-key frames. Also, neighboring frames are highly correlated which are also utilized to compensate the next frame. *Fig3.8* depicts the flow of Deep reconstruction for key frame.

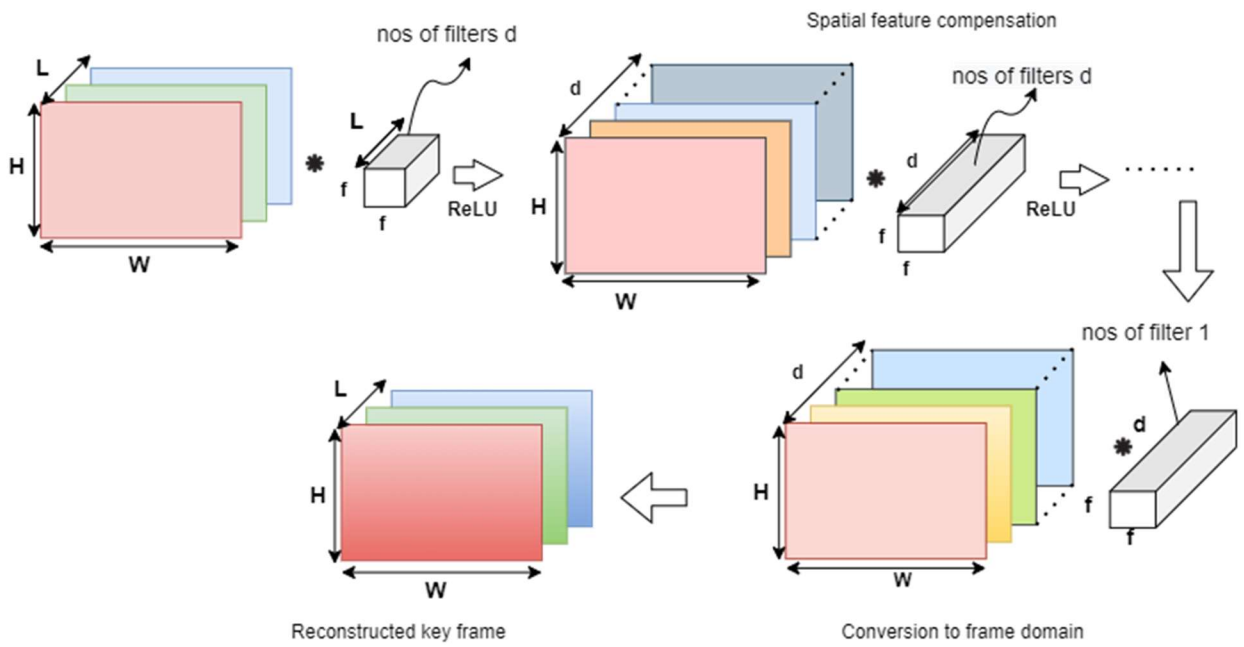

Fig3.8: Restoration of the key frame in deep network by spatial feature compensation

Mathematically this reconstruction can be given as,

$$DR^{1}_{key}(\check{X}_{key}) = \ ReLU(W^{1}_{dr,key} * \check{X}_{key} \ + \ b^{1}_{key} \ ) \tag{3.7}$$

$$DR^{i}_{key}(\check{X}_{key}) = \ ReLU(W^{i}_{dr,key} * DR^{i-1}_{key}(\check{X}_{key}) \ + \ b^{i}_{key} \quad i = 2,3..n \tag{3.8}$$

$$DR^{pred}_{key}(\check{X}_{key}) = \ W^{pred}_{dr,key} * DR^{n}_{key}(\check{X}_{key}) \ + \ b^{pred}_{key} \tag{3.9}$$

Where $ReLU$ is a simple activation function that is used to visualize features of individual blocks in deep network.

For (3.7) the filter of the first stage of the deep reconstruction network $(W^{1}_{dr,key})$ has d filters of size $f \times f \times L$. and the later stages for $(i = 2,3 \ldots n)$ the filters of the network consist of d filters of size $f \times f \times d$ as there are d features are exploited for spatial feature compensation. In the final stage filter $(W^{pred}_{dr,key})$ of the deep

network consist of L filters of size $f \times f \times d$. The $\boldsymbol{b}$ term is for the bias term for the respective layers.

The non-key frame network follows the same feed forward structure of the key frame deep reconstruction network but it additionally has deep feature compensation form the key frame and we have included the deep feature compensation from the neighboring frame in the pre-final deep reconstruction block.

For each non-key frame except for the frames adjacent to key frame the network equation can be formulated as bellow,

Mathematically this reconstruction can be given as,

$$DR_{nk}^{1}(\check{X}_{nk}) = \ ReLU(W_{dr,nk}^{1} * \check{X}_{nk} + \ W_{ref,key}^{1} * DR_{key}^{1}(\check{X}_{key}) + \ b_{nk}^{1} \ ) \quad (3.10)$$

$$DR_{nk}^{i}(\check{X}_{nk}) = \ ReLU\big(W_{dr,nk}^{i} * DR_{nk}^{i-1}(\check{X}_{nk}) + \ W_{dr,key}^{i} * DR_{key}^{i}(\check{X}_{key}) + b_{dr,key}^{i} \ + \ b_{nk}^{i} \ \big) \quad (3.11)$$

$$where\ i = 2,3..n-1$$

$$DR_{nk}^{n}(\check{X}_{nk}) = \ ReLU\big(W_{dr,nk}^{n} * DR_{nk}^{n-1}(\check{X}_{nk}) + W_{ref,key}^{n} * DR_{key}^{n}(\check{X}_{key}) \ + W_{ref,nk\therefore 1}^{n} \ * DR_{key}^{1}(\check{X}_{nk\therefore 1}) + \ b_{nk-1}^{n} + \ b_{dr,key}^{n} + \ b_{refnk\therefore 1}^{n}\big) \quad (3.12)$$

$$DR_{nk}^{pred}(\check{X}_{nk}) = \ W_{dr,nk}^{pred} * DR_{nk}^{n}(\check{X}_{nk}) \ + \ b_{nk}^{pred} \quad (3.13)$$

Since we use the middle frame of the GOP as the key frame son the compensation od deep feature follows the *Fig3.4*. So, the $\therefore$ sign is used in (3.12) which is replaced by subtraction $(-)$ for the frame whose index is higher than the key frame and the sign is replaced by $(+)$ for frame reconstruction network of the frames having lower index no than the key frame.

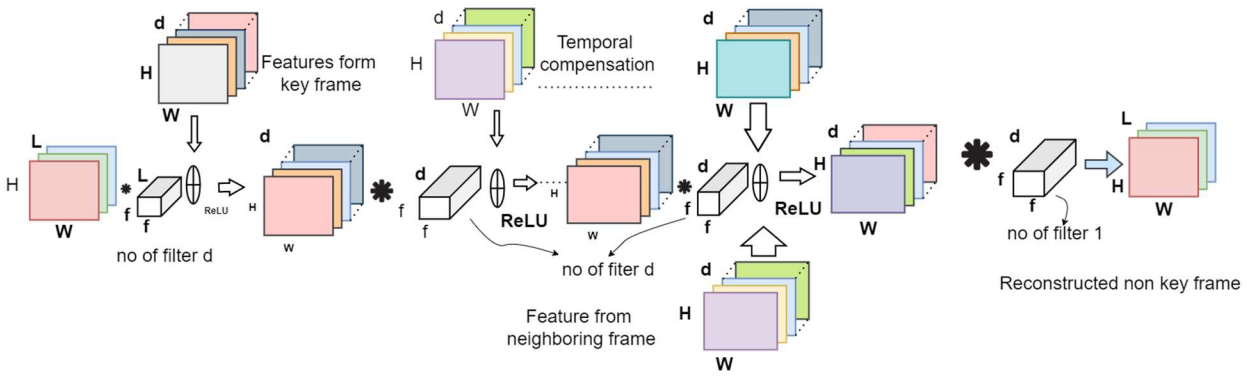

Fig3.9: Elaborate description of reconstruction of non-key frame by spatial and temporal compensation in deep reconstruction

Fig3.9 shows the restoration of non-key frame which are not adjacent to the key frame. For the adjacent frames of the key frame since there is no previous non-key frame to compensation from the only temporal compensation is done from the key frame. The network equation can be mathematically represented as described in (3.14), (3.15), (3,16).

$$DR_{nk}^{1}(\check{X}_{nk}) = \ ReLU(W_{dr,nk}^{1} * \check{X}_{nk} + \ W_{ref,key}^{1} * DR_{key}^{1}(\check{X}_{key}) + \ b_{nk}^{1} \ ) \quad (3.14)$$

$$DR_{nk}^{i}(\check{X}_{nk}) = \ ReLU(W_{dr,nk}^{i} * DR_{nk}^{i-1}(\check{X}_{nk}) + \ W_{dr,key}^{i} * DR_{key}^{i}(\check{X}_{key}) + \ b_{dr,key}^{i} \ + \ b_{nk}^{i} \ ) \quad (3.15)$$

$$where\ i = 2,3 \ldots . n$$

$$DR_{nk}^{pred}(\check{X}_{nk}) = \ W_{dr,nk}^{pred} * DR_{nk}^{n}(\check{X}_{nk}) \ + \ b_{nk}^{pred} \quad (3.16)$$

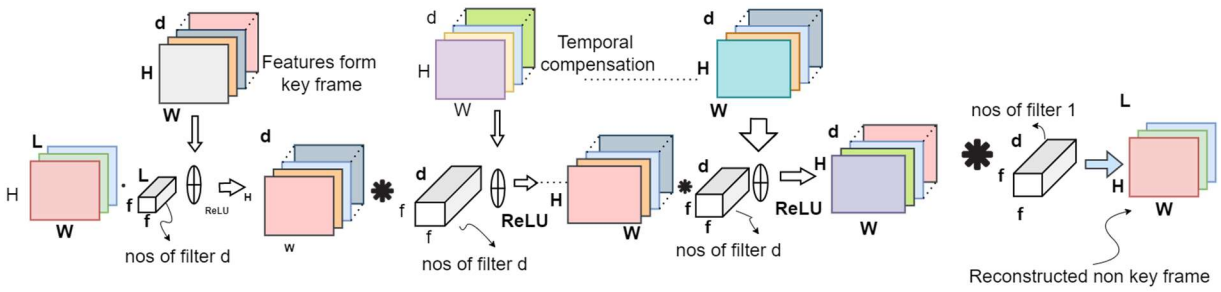

Fig3.10: Elaborate description of reconstruction of non-key frame that are adjacent to key frame by spatial and temporal compensation in deep reconstruction

The feed forward network filters are of similar size as of feed forward network of key frame, i.e. $W^{1}_{dr,nk}$ has d filters of support $f \times f \times L$ and the intermediate stages ($W^{i}_{dr,nk}$) has d filters of support $f \times f \times d$ and the final prediction block has L filters of support $f \times f \times d$. Also, deep feature compensation layers have from both key and previous non-key frame ($W^{i}_{dr,key}, W^{n}_{re\ ,nk:1}\ respectively$) has d filters of size $f \times f \times d$.

3.5. Training:

In many works a random collection of elements from earlier works is frequently used to handle the sampling matrix. In our case, a convolution layer is used to construct block-based framewise sampling, allowing it to be learned flexibly. Also, since the sampling layer's output is the initial reconstruction's input, and the initial reconstruction's output is the deep reconstruction's input. All these three components of our proposed network learned in conjunction while training.

For optimizing the network, for non-key frames we have considered the prediction output of the network as well as the initial frame level restoration of the frame is our loss function by comparing these with the original frames. For key frame only, the

final prediction is considered in loss function. As a result, total loss of the network consists of $2K + 1$ terms for K non-key frame and one key frame.

Also, since The Structural Similarity Index based loss function performs better than the Euclidian loss function or the mean squire error(MSE) for image restoration [the ssim paper] we have considered it as a metric of restored image quality in comparison with original frame.

The loss function for a restored frame(D_k) and original frame(X_k) using $SSIM$ is given as,

$$ESSIM(X_k, D_k) = 1 - \frac{1 + SSIM(D_k, X_k)}{2} \tag{3.17}$$

Where $ESSIM(\cdot)$ error or by calculating structural dissimilarities between restored frame(D_k) and original frame(X_k).

So, the total loss for a training data of N GOPs consisting of K non-key frame and single key frame is given by,

$$loss(\theta) = \sum_{i=1}^{N} \left(\left| ESSIM\ (X_{key}, D_{key}) \right| + \sum_{nk=1}^{K} \left| ESSIM(X_{nk}^{(i)}, D_{nk}^{(i)}) \right| + \left| ESSIM(X_{nk}^{(i)}, I_{nk}^{(i)}) \right| \right) \tag{3.18}$$

For the back propagation in the training time the derivative output of the loss function is required which is calculated by the following method.

To calculative derivative of the loss term $ESSIM(\cdot)$ we get that it is directly proportional to negative of $SSIM$ *i.e.* $\propto -SSIM$.

$SSIM$ for a pixel m for image x and y is given by,

$$SSIM(p) = \frac{\mu_x\mu_y + C_1}{\mu_x + \mu_y + C_1} \cdot \frac{\sigma_{xy} + C_2}{\sigma_x + \sigma_y + C_2}$$
$$= l(m) \cdot cs(m) \quad (3.19)$$

Here, μ , σ bears usual meaning and $C's$ are constant.

Now, to calculate derivative Remember that we have to calculate the derivatives at m with respect to any other pixel n in patch$P$. [ssim paper]

$$\frac{\partial\, SSIM}{\partial\, x(n)} = \left(\frac{\partial\, l(m)}{\partial x(n)} \cdot cs(m) + l(m) \cdot \frac{\partial\, cs(m)}{\partial x(n)}\right) \quad (3.20)$$

The derivative of the of the terms of $SSIM$ is given as,

$$\frac{\partial\, \mathrm{l(m)}}{\partial \mathrm{x(n)}} = 2 \cdot G_{\sigma_G}(n-m) \cdot \left(\frac{\mu_x - \mu_y \cdot l(m)}{\mu_x^2 + \mu_y^2 + C_1}\right) \quad (3.21)$$

$$\frac{\partial\, \mathrm{cs(m)}}{\partial \mathrm{x(n)}} = \frac{2}{\sigma_x^2 + \sigma_y^2 + C_2} \cdot G_{\sigma_G}(n-m)$$
$$\cdot \left[(y(n) - \mu_y) - cs(m) \cdot (x(n) - \mu_x)\right] \quad (3.22)$$

Here $G_{\sigma_G}(n-m)$ is Gaussian coefficient, which is related to the pixel n.

These two terms are used in (3.20) to calculate derivative of $SSIM$ which is done all over the image frame to obtain the derivative for all the pixel. This way derivative output from $ESSIM(\cdot)$ is obtained.

We know that transfer learning is used improve performance of a deep network in training stage as it allows the network to converge the weights and biases of the network to the optimal value is short time. So, we have pretrained our model's feed forward connection by using the optimal weights of CSnet [csnet] and for the multi-level feature compensation we have used the weights of VCSnet[cnn]. We kept the learning rate of key frame's feed forward network component to zero and another

network component as one. Our model was built using MatConvNet [28] library in MATLAB by using DagNN wrapper.

$Adaptive\ moment\ estimation$ ($ADAM$) [29] was as the optimizer of our network parameters.

CHAPTER 4: EXPERIMENTAL RESULTS

4.1. Experimental setup:

The training data set is formed using two standard HEVC video sequence $Kimino$ and $Cactus$ by dividing them into GOPs consisting of 8 frames each having size 96×96. These videos have resolution of 1920×1080. So, consecutive 8 frames are taken to create patch GOPs of size $98 \times 96 \times 8$ by taking 8 patches having same indices in their respective frames. All these patches were also augmented four times simultaneously (for keeping the motion information in the neighboring frame intact) for one GOP. Stride of 60×60 was taken while traversing one frame the frames in 96×96 size. As a result, we got a dataset for training which consist of $1{,}10000$ numbers of GOPs. The validation dataset was similarly formed by taking 12 GOPs form two standard HEVC video sequence $Ready$ and $Basket$ each. They are also of resolution 1920×1080. Similarly like training dataset this validation dataset also created by taking patches of size 96×96 from eight consecutive frames and augmented two times and the stride

was same as before 60×60.

For test our model we have used six test video sequence named $Akiyo, Coastguard, Foreman, Mother\ daughter, Paris, Silent$ which has resolution of 96×96.

In the framewise sampling using block-compressed-sensing in our method, the block size is set to $32 \times 32 \times 1$. During framewise initial reconstruction the filter size is adjusted adaptively based on the sampling ratio. We used $f = 3, L = 1, d = 64$ and $n = 4$ in the deep restoration network for multilevel feature compensation.

We investigate our proposed method for two sampling ratios 0.1 and 0.01.

4.2. *Result for 0.01 sub rate:*

For 0.01 sub-rate we have trained our network by above database for 20 epochs with learning rate 0.01. We have got the following *training loss* vs. *validation loss* graph (*Fig4.1*).

Sequence name	***Frame index***							
	1	***2***	***3***	***4***	***5***	***6***	***7***	***8***
Akiyo	32.77	31.96	34.08	44.2	33.56	31.76	31.69	31.15
Coast guard	24.25	24.98	27.41	39.41	26.93	24.73	24.06	23.73
Foreman	27.19	27.88	29.15	39.7	28.76	27.53	27.39	27.49
Mother_daughter	33.56	34.53	35.41	47.16	35.44	33.83	32.79	32.17
Paris	22.54	22.12	24.1	31.67	23.96	22.38	22.05	21.39
Silent	30.02	29.9	30.86	38.67	30.46	27.9	28.26	28.23

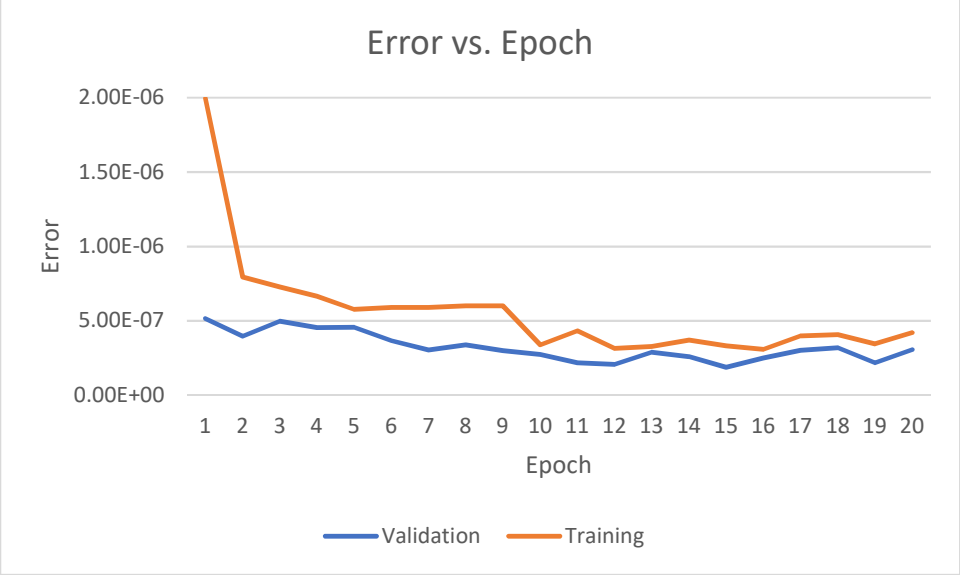

Fig 4.1: Training vs validation error for different epochs for sub-rate 0.01

Using this network, we achieved the results for the first 8 frames of each of the six test video sequences shown in the ***TABLE I*** below.

Table I. *PSNR of obtained prediction for first GOP of each testing data at sub-rate 0.01*

For the second GOP *PSNR* of obtained results are Shown in ***TABLE II***.

Sequence name	Frame index							
	9	***10***	***11***	***12***	***13***	***14***	***15***	***16***
Akiyo	32.33	31.77	33.92	44.19	33.51	31.67	31.41	30.61
Coast guard	23.95	24.75	27.14	39.21	26.57	24.51	23.67	23.22
Foreman	24.46	25.33	27.79	39.68	27.48	25.53	24.79	24.07
Mother_daughter	34.08	34.95	35.59	47.18	35.50	34.31	33.90	33.46
Paris	22.17	21.93	24.05	31.76	23.88	22.28	21.99	21.42
Silent	28.91	29.00	29.95	38.79	29.79	27.46	27.65	27.37

Table II. *PSNR of obtained prediction for second GOP of each testing data at sub-rate 0.01*

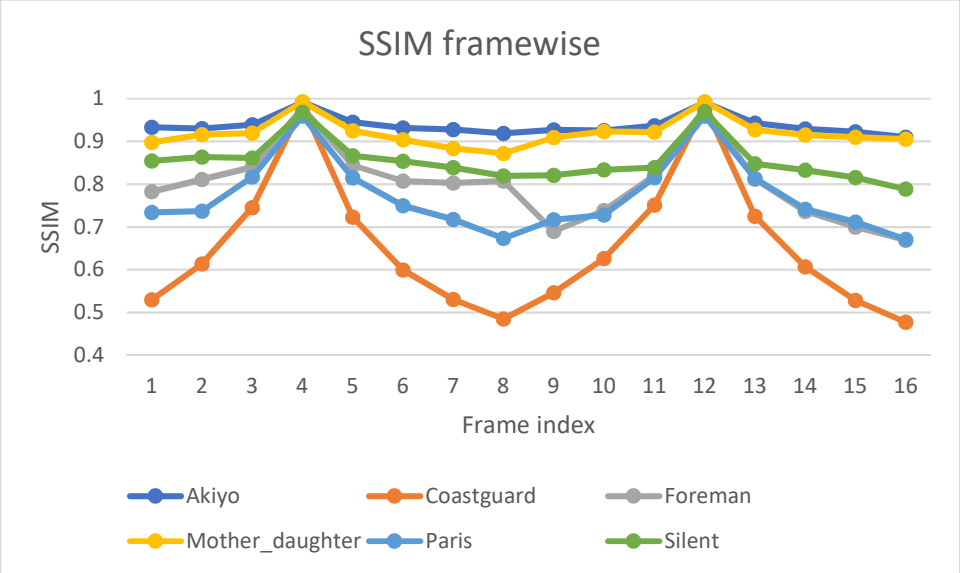

Fig4.2: Comparison of SSIM of different frames for sub rate 0.01

Reconstructed video frames (2nd to 6th frames) are shown in *Fig4.3.* for SR 0.01.

$(a)\ Akiyo$

$(b)\ Coastguard$

$(c)\ Foreman$

$(d) Mother\ Daughter$

$(e) Paris$

(f) *Silent*

Fig4.3: Reconstructed video frames (from 2nd to 6th frame) for all six test videos (a) Akiyo (b) Coastguard (c) Foreman (d) Mother Daughter (e) Paris (f) Silent for sub rate 0.01.

4.3. Results for 0.1 sub rate:

During the training of the network dedicated for 0.1 sub-rate we have obtained the following $training\ loss$ vs.$validation\ loss\ upto$ 47 epochs. Where the learning rate was 0.01 for the initial 16 epoch and for the later epochs the learning rate was 0.001. The obtained data is given in *Fig4.4.*

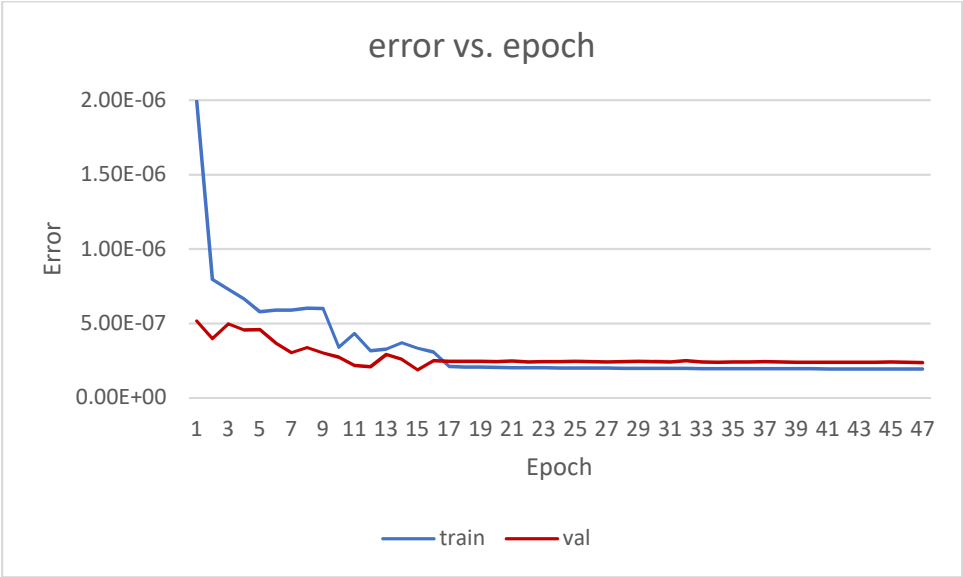

Fig4.4: Training vs validation error for sub rate 0.1 up to 47 epochs

Below in ***TABLE III*** we have shown result obtained for different epochs for 0.1 sub-rate and have shown how gradually the results have improved.

Sequence name	***Epoch no***									
	15		***16***		***20***		***34***		***47***	
	psnr	*ssim*	*psnr*	*ssim*	*psnr*	*ssim*	*psnr*	*ssim*	*psnr*	*ssim*
Akiyo	37.03	0.96	36.76	0.96	35.82	0.95	39.71	0.98	40.11	0.98
Coast guard	29.22	0.75	29.22	0.75	29.01	0.75	29.34	0.76	29.33	0.75
Foreman	32.22	0.89	31.96	0.89	31.56	0.89	33.28	0.91	33.37	0.91
Mother_daughter	38.4	0.94	37.88	0.94	37.05	0.94	40.47	0.96	40.75	0.96
Paris	26.04	0.84	26.05	0.84	25.72	0.83	27.11	0.88	27.25	0.88
Silent	33.41	0.89	33.18	0.89	32.58	0.89	35.15	0.93	35.36	0.93
Avg.	32.72	0.88	32.51	0.88	31.96	0.88	34.18	0.90	34.36	0.90

Table III. *Improvement of PSNR and SSIM as increase of epoch for sub rate 0.1*

In *Fig4.5* we have shown gradual improvement of the of $average\ PSNR$ of the recovered test video sequence.

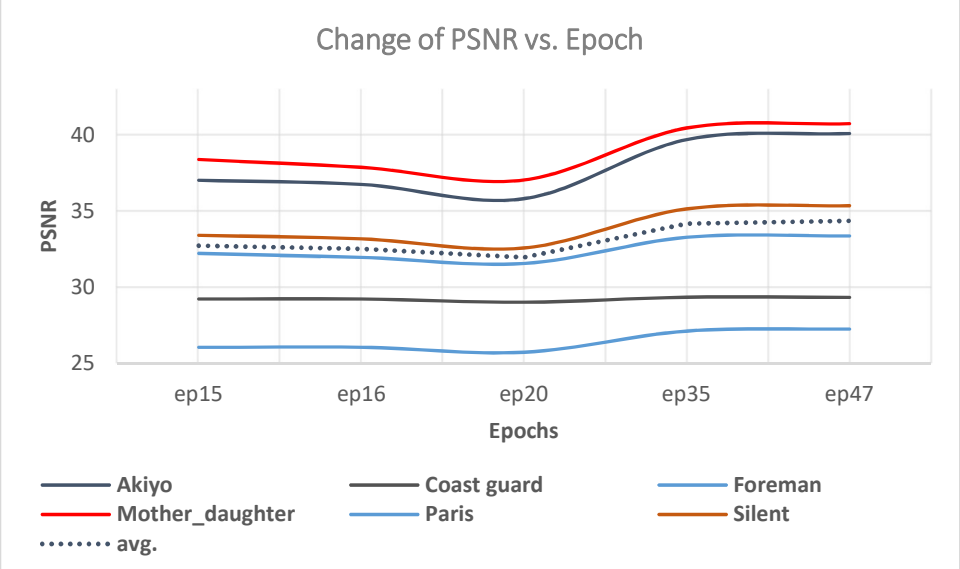

Fig4.5: Gradual improvement of average PSNR of Recovered test video sequence

Table IV and ***Table V*** provides *PSNR* obtained for all restored video frames after using the network after training for 47 epochs for sub-rate 0.1.

Sequence Name	***Frame Index***								***Avg***
	1	***2***	***3***	***4***	***5***	***6***	***7***	***8***	
Akiyo	39.59	40.98	40.84	44.20	40.00	40.16	39.48	38.55	**40.47**
Coast guard	27.42	27.85	29.45	39.41	29.06	27.66	27.32	27.19	**29.42**
Foreman	31.93	33.37	34.90	39.70	34.28	32.74	32.40	33.22	**34.07**
Mother_daughter	38.42	40.09	41.62	47.16	40.38	39.19	38.42	38.20	**40.43**
Paris	26.32	27.04	27.50	31.67	27.25	26.82	26.01	25.63	**27.28**
Silent	35.03	36.01	36.08	38.67	35.62	34.79	34.57	34.15	**35.61**

Table IV. *PSNR of obtained prediction for first GOP of each testing video at sub-rate 0.1*

Sequence Name	***Frame Index***								***Avg***
	9	***10***	***11***	***12***	***13***	***14***	***15***	***16***	
Akiyo	39.04	40.41	40.55	44.19	39.71	39.54	38.06	36.53	**39.75**
Coast guard	27.27	27.72	29.37	39.21	29.05	27.56	26.98	26.73	**29.24**
Foreman	31.38	31.74	32.71	39.68	32.91	31.39	30.82	30.80	**32.68**
Mother_daughter	39.07	41.02	41.87	47.18	40.47	39.74	39.48	39.64	**41.06**
Paris	26.05	26.72	27.40	31.76	27.17	26.78	26.11	25.79	**27.22**
Silent	34.16	34.85	35.41	38.79	35.19	34.31	34.24	33.93	**35.11**

Table V. *PSNR of obtained prediction for second GOP of each testing video at sub-rate 0.1*

Change of Structural similarity as the distance from key frame increases is shown below for all the test sequence for their first two GOPs compressed with sub-rate 0.1 in *Fig4.6*.

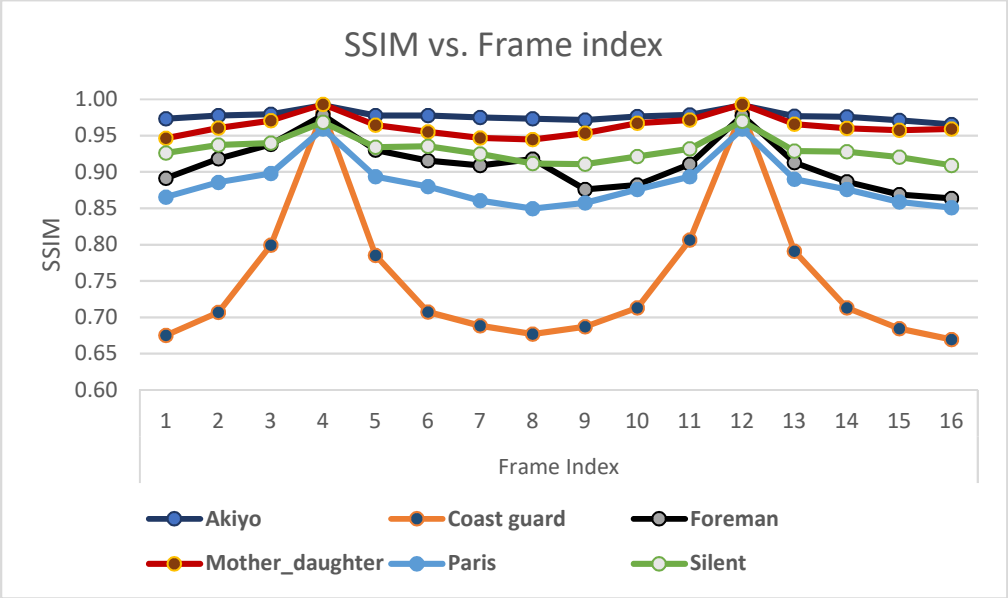

Fig4.6: SSIM for individual frames of the recovered test sequence for sub rate 0.1

The reconstructed frames (2nd to 6th) for all the test video sequence are shown in *Fig4.7* for sub-rate 0.1.

(a) *Akiyo*

(b) *Coastguard*

(c) *Foreman*

(d)*Mother Daughter*

(e) *Paris*

(f) *Silent*

Fig4.7. Reconstructed video frames of six test video sequences (a) Akiyo, (b) Coastguard, (c) Foreman, (d)Mother daughter (e) Paris (f) Silent for sub-rate 0.1

4.4. Comparison:

Table VI shows the comparison between the existing deep-learning based video reconstruction method for compressive sensing video that only uses one key frame for video reconstruction.

Sequence name	***Sub-rate***	***Ours***	***VCSnet1***	***ISTA-Net+***	***CSnet***
Akiyo	0.01	**33.79**	33.67	25.12	29.03
Coast guard		**26.78**	26.64	21.85	25.04
Foreman		**28.39**	27.57	22.27	26.03
Mother_daughter		**35.87**	35.8	24.93	30.33
Paris		**23.73**	23.26	19.15	20.54
Silent		30.20	**30.52**	23.11	26
Avg.		**29.79**	29.58	22.74	26.16
Akiyo	0.1	**40.11**	39.72	34.83	35.36
Coast guard		29.33	**29.38**	27.23	29.13
Foreman		33.37	33.4	32.83	32.38
Mother_daughter		40.75	**40.88**	35.54	37.18
Paris		**27.25**	26.58	24.07	24.66
Silent		35.36	**35.78**	30.23	31.82
Avg.		**34.36**	34.29	30.79	31.76

Table VI. *Comparison of average PSNR of obtained prediction for all the test videos for sub-rate 0.1 and 0.01*

Our obtained results beat the results obtained in VCSnet1 [25] for single key frame for five out of six test video sequences for the sub-rate 0.01 and gives very good results for sub-rate 0.1 which is very close to the results obtained for VCSnet1.

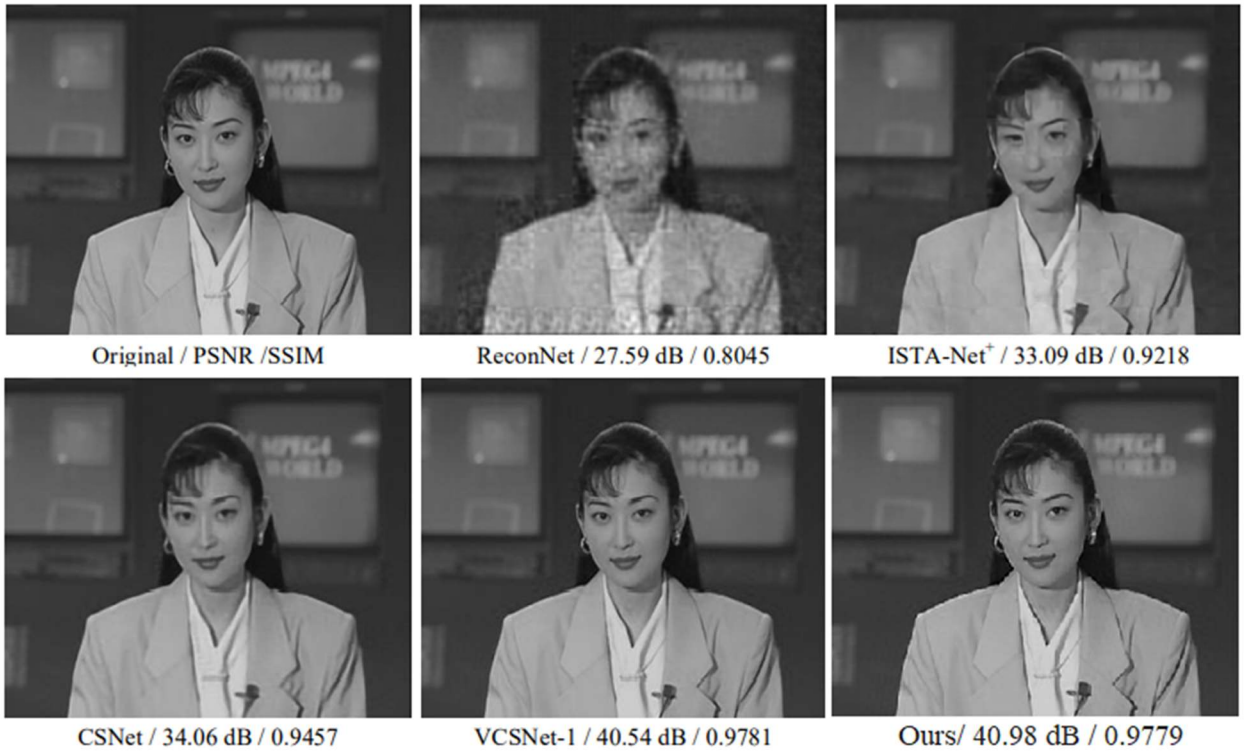

Fig4.8: Comparison of 2nd frame of the restored video for different deep-learning method for sub-rate 0.1

Our method also performs better in comparison with other deep-learning based restoration methods such as ReconNet [19], ISTA-Net+ [22], CSNet [20] for both 0.1 and 0.01 sub-rate.

Fig4.9 gives the comparison of result obtained by our method and the other available deep learning based available for video/image recovery.

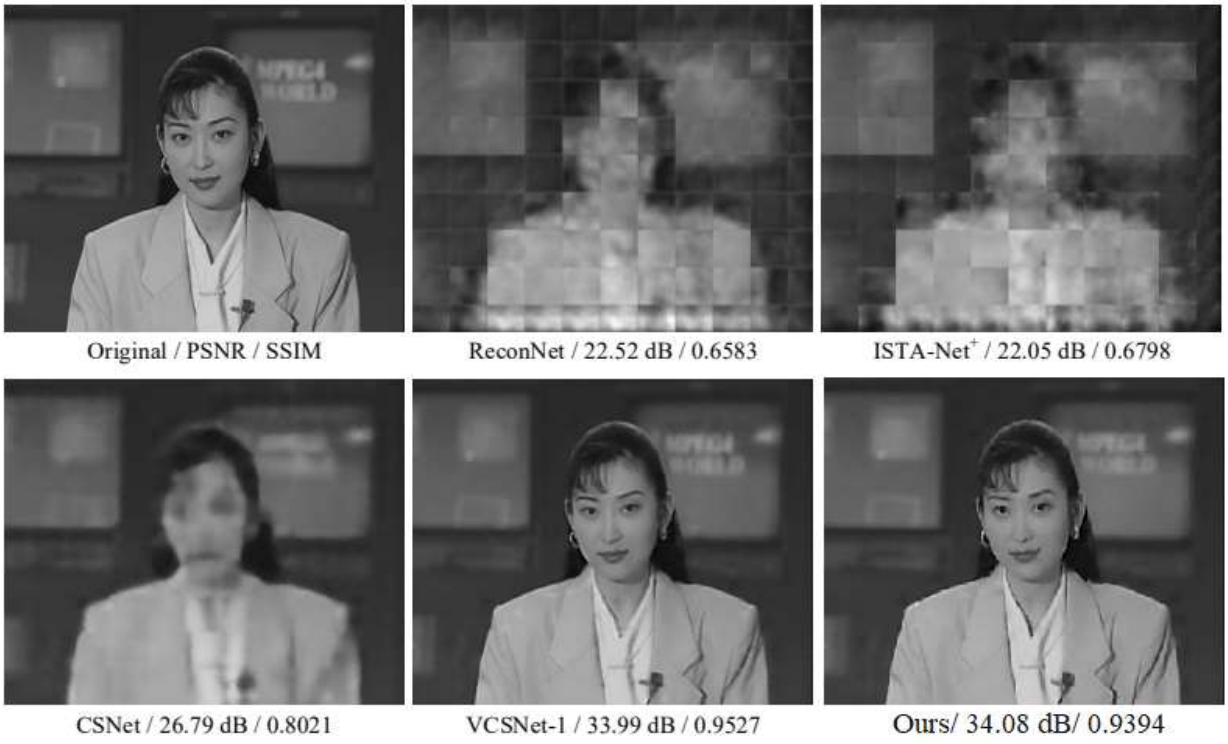

Fig4.9: Comparison between neighboring video frames of key frame for all the deep-learning method operating with sub-rate 0.01

From *Fig4.9* we can see that our method that uses single key and previous non-key frame of reconstruction of video frames out perform VCSnet1 which used single key frame in deep restoration process by 0.09 dB after training for only 20 epochs for sub rate of 0.01. It also provides much better restored quality than the other deep methods of compressed sensed video restoration.

CHAPTER 5: CONCLUSION

In this paper we have discussed a convolutional neural network based on deep network for video compressive sensing and restoration using deep spatial and temporal data from key frame and neighboring non-key frame. Also, we have introduced $SSIM$ based loss function increasing the quality of the restored frames instead of MSE. We improved the security of the transmitter network for data privacy by inserting a novel classical encryption module based on chaotic sequence and MD's matrix-based diffusion. We have also compared our method with other classical and deep-learning based method and have found that our method performs better than these available methods. But our network performs lesser than the state-of-the-art method VCSnet2 which uses double key frame for temporal compensation in their deep restoration network. So, if we also take help of double key frame there is a significant chance that our method may outperform VCSnet2. Also, inclusion of multi-scale SSIM as a loss function in video frame restoration can give a significant increase in reconstructed video quality.

5. CHAPTER 6: REFERENCES

[1]. Donoho David L.," Compressed sensing." Inform. Theory, IEEE Trans. on, 52.4 (2006): 1289-1306.

[2]. E. J. Candès. Compressive sensing. Proceedings of the International Congress of Mathematics, 2006.

[3]. R. G. Baraniuk. Compressive sensing. IEEE Signal Processing Magazine, 24(4):118–120, 2007

[4]. J. Tropp and A. Gilbert, "Signal recovery from random measurements via orthogonal matching pursuit," IEEE Transactions on Information Theory, vol. 53, no. 12, pp. 4655–4666, December 2007.

[5]. L. Gan, "Block compressed sensing of natural images," in Proceedings of the International Conference on Digital Signal Processing, Cardiff, UK, July 2007, pp. 403–406.

[6]. S. Mun and J. E. Fowler, "Residual reconstruction for block-based compressed sensing of video," in Proceedings of the IEEE Data Compression Conference, J. A. Storer and M. W. Marcellin, Eds., Snowbird, UT, March 2011.

[7]. M. B. Wakin, J. N. Laska, M. F. Duarte, D. Baron, S. Sarvotham, D. Takhar, K. F. Kelly, and R. G. Baraniuk, "An architecture for compressive imaging," in Proceedings of the International Conference on Image Processing, Atlanta, GA, October 2006, pp. 1273–1276.

[8]. "Compressive imaging for video representation and coding," in Proceedings of the Picture Coding Symposium, Beijing, China, April 2006.

[9]. "Residual reconstruction for block-based compressed sensing of video," in Proceedings of the IEEE Data Compression Conference, J. A. Storer and M. W. Marcellin, Eds., Snowbird, UT, March 2011, pp. 183–192.

[10]. G. J. Sullivan, "Multi-hypothesis motion compensation for low bit-rate video coding," in Proceedings of the International Conference on Acoustics, Speech, and Signal Processing, vol. 5, Minneapolis, MN, April 1993, pp. 437–440.

[11]. "Efficiency analysis of multihypothesis motion-compensated prediction for video coding," IEEE Transactions on Image Processing, vol. 9, no. 2, pp. 173–183, February 2000.

[12]. Li R, Liu H, Xue R, Li Y. Compressive-Sensing-Based Video Codec by Autoregressive Prediction and Adaptive Residual Recovery. International Journal of Distributed Sensor Networks. August 2015. doi:10.1155/2015/562840

[13]. Chen, J., Wang, N., Xue, F. *et al.* Distributed compressed video sensing based on the optimization of hypothesis set update technique. *Multimed Tools Appl* **76,** 15735–15754 (2017).

[14]. Chen, Can & Zhou, Chao & Liu, Pengyuan & Zhang, Dengyin. (2018). Iterative Reweighted Tikhonov-Regularized Multihypothesis Prediction Scheme for Distributed Compressive Video Sensing. IEEE Transactions on Circuits and Systems for Video Technology. PP. 1-1. 10.1109/TCSVT.2018.2886310.

[15]. C. Zhao, S. Ma, J. Zhang, R. Xiong and W. Gao, "Video Compressive Sensing Reconstruction via Reweighted Residual Sparsity," in IEEE Transactions on Circuits and Systems for Video Technology, vol. 27, no. 6, pp. 1182-1195, June 2017, doi: 10.1109/TCSVT.2016.2527181.C.

[16]. Li, W. Yin, and Y. Zhang, "Users guide for TVAL3: TV minimization by augmented lagrangian and alternating direction algorithms," CAAM report, 2009.

[17]. S. Mun and J. E. Fowler, "Block compressed sensing of images using directional transforms," in IEEE International Conference on Image Processing (ICIP), 2009, pp. 3021–3024.

[18]. A. C. Sankaranarayanan, C. Studer, and R. G. Baraniuk, "CS-MUVI: Video compressive sensing for spatial-multiplexing cameras," in IEEE International Conference on Computational Photography (ICCP), 2012, pp. 1–10.

[19]. Y. T. Yang, J. Sun, H. Li, and Z. Xu, "Deep admm-net for compressive sensing mri," in Advances in neural information processing systems, 2016, pp. 10–18.

[20]. S. Boyd, N. Parikh, E. Chu, B. Peleato, and J. Eckstein, "Distributed optimization and statistical learning via the alternating direction method of multipliers," Foundation and Trends in Machine Learning, vol. 3, no. 1, pp. 1–122, 2011.

[21]. J. Zhang and B. Ghanem, “Ista-net: Interpretable optimization-inspired deep network for image compressive sensing,” in Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition, 2018, pp. 1828–1837.

[22]. K. Kulkarni, S. Lohit, P. Turaga, R. Kerviche, and A. Ashok, “Recon1net: Non-iterative reconstruction of images from compressively sensed measurements,” in Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition, 2016, pp. 449–458.

[23]. K. Xu, Z. Zhang, and F. Ren, “Lapran: A scalable Laplacian pyramid reconstructive adversarial network for flexible compressive sensing reconstruction,” in European Conference on Computer Vision. Springer, 2018, pp. 491–507.

[24]. W. Shi, F. Jiang, S. Liu, and D. Zhao, “Scalable convolutional neural network for image compressed sensing,” in Computer Vision and Pattern Recognition, 2019.

[25]. W. Shi, S. Liu, F. Jiang and D. Zhao, "Video Compressed Sensing Using a Convolutional Neural Network," in IEEE Transactions on Circuits and Systems for Video Technology, vol. 31, no. 2, pp. 425-438, Feb. 2021, doi: 10.1109/TCSVT.2020.2978703.

[26]. Steven L. Brunton and J. Nathan Kutz, *Data-Driven Science and Engineering: Machine Learning, Dynamical Systems, and Control* University Printing House, Cambridge CB2 8BS, United Kingdom.

[27]. A. N. Tikhonov and V. A. Arsenin. Solution of Ill-posed Problems. Winston & Sons, Washington, (1977).

[28]. A. Vedaldi and K. Lenc, “Matconvnet: Convolutional neural networks for matlab,” in Proceedings of the 23rd ACM International Conference on Multimedia. ACM, 2015, pp. 689–692.

[29]. D. Kingma and J. Ba, “Adam: A method for stochastic optimization,” arXiv preprint arXiv:1412.6980, 2014