CHAPTER 4
SUPER - RESOLUTION

4.1 INTRODUCTION

Now a days, the smart phones have camera with 4K ultra HD display for recording. This will be one of the main factors for buying the smart phones. The 4K resolution means 4096x2160 pixels. Many 4K DSLRs (Digital single – lens reflex camera), camcorders and video recorders capture the video about 24 to 30 frames per second. Recently, iPhone 6s Plus has been used to do the job of the DSLRs i.e., used in film documentaries. The positive points of using the smart phones are that

- Everyone will carry the exquisite video quality in their pocket, i.e., more compact
- The image details in the photos and videos are extraordinary
- Creates high quality videos and images in a perfect manner
- Bragging power
- Professional videos in just a mobile phone

In the past, there were professional photographers to shoot the video of memorable events like betrothal, wedding, reception, birthday party, any social functions etc., But now a days, everyone has higher end mobile phones and takes the videos. This is welcoming one because, everyone is fond of photography. The problems with such kind of mobile phones are

- Expensive
- Some of mobile phones can quickly over heat while taking videos
- The life of the battery is seriously affected
• Even though 4K resolution, it displays in a tiny 5 to 6 inch screen

Even though the 4K smart phones come with their benefits, every one cannot afford to buy such phones. Thus, resolution plays a vital role in defining the quality of the video. Hence, the present research work has been concentrated on producing high resolution image sequences from low resolution image sequences. There are plenty of single frame super-resolution algorithms available. By the use of such algorithms, the low resolution video sequences are converted to high resolution image sequences.

4.2 STRUCTURE MODULATED SPARSE REPRESENTATION METHOD

In this method (Yongqin Zhang et al 2015), the input low resolution images have been initially enhanced based on Ridge regression. Up sampling and down sampling are employed to enhance the image. The noises occurring due to the up sampling and down sampling of the images are then handled based on the optimization using the solution of regression problem. The noises in the optimized images are then removed using Non local means filter. Sparse coding is then applied to the image and then, convex minimization problem is handled. Finally, reconstructed high resolution image is obtained. The flow chart is given in the figure4.1. The image gradient histogram of a LR input is incorporated as a gradient regularization term in the image sparse representation model. This SMSR algorithm employs the gradient prior and non-locally centralized sparsity to design the constrained optimization problem for training the dictionary and HR image reconstruction. The first stage output of the SMSR method is made up of two stages: the gradual magnification and the structured sparse representation. The system has proposed a joint super-resolution framework of structure-modulated sparse representations to improve the performance of sparsity-based image super-resolution. The algorithm formulates the constrained optimization problem for high-resolution image recovery. The multistep magnification
scheme with ridge regression is first used to exploit the multiscale redundancy for the initial estimation of the high-resolution image. Then, the gradient histogram preservation is incorporated as a regularization term in sparse modeling of the image super-resolution problem. Finally, the numerical solution is provided to solve the super-resolution problem of model parameter estimation and sparse representation.

![Image of Architecture of SMSR algorithm]

**Figure 4.1 Architecture of SMSR algorithm**

The reason for selecting this method is that it is suitable for all the types of images. The proposed method helps in identifying the accurate blur
kernel in the images and hence, the visual similarities in the images are achieved more effectively. The original color informations are well preserved in the resulting image. The upsampling and downsampling will make the scaling of the images in different levels and also in each levels, the original pixel informations are well preserved. The identification of the noise levels and the usage of sparse matrix improve the overall performance of the process.

4.2.1 Ridge Regression

The input low resolution images are obtained from the dataset. In the ridge regression step, gradual magnification of the images is done. The values of a given pixel in the output image are calculated by multiplying each kernel value by the corresponding input image pixel values. If kernel is symmetric, place the center (origin) of kernel on the current pixel. Then, the kernel will be overlapped with neighboring pixels too. Now, multiply each kernel element with the pixel value that it is overlapped with and add all the obtained values. Resultant value will be the value for the current pixel that is overlapped with the center of the kernel. The convolution of blur in the images produces blur in the images and it is helpful in preserving original pixel information in the images. The flowchart of ridge regression is given in the figure 4.2 and the corresponding output obtained for butterfly image is given in figure 4.3.

If the kernel is not symmetric, it has to be flipped both around its horizontal and vertical axes before calculating the convolution. The Gaussian kernel is the physical equivalent of the mathematical point. It is not strictly local, like the mathematical point, but semi-local. It has a Gaussian weighted extent, indicated by its inner scales. Because, scale-space theory is revolving around the gaussian function and its derivatives are considered as a physical differential operator i.e. the mathematical notions underly sampling of values from functions and their derivatives at selected points (i.e. that is why it is referred to as sampling). The mathematical functions involved are the generalized functions,
i.e. the Delta-Dirac function, the Heavyside function and the error function. The normalized Gaussian kernel has an area under the curve of unity, i.e. as a filter, it does not multiply the operand with an accidental multiplication factor. Two Gaussian functions can be cascaded, i.e. applied consecutively, to give a Gaussian convolution result which is equivalent to a kernel with the variance equal to the sum of the variances of the constituting Gaussian kernels. The spatial parameter normalized over scale is called the dimensionless ‘natural coordinate’.

![Diagram](image.png)

**Figure 4.2 Ridge regression**

![Image](image.png)

**Figure 4.3 Result of added isotropic kernel**
4.2.2 Up Sampling and Down Sampling

The images are up sampled and down sampled based on bicubic interpolation. The steps involved is given in figure 4.4. In bicubic interpolation based on the size of the magnification scale, the rows and columns of the images are updated. The input image rows are replaced in the neighboring rows so that, the image is enhanced in the row fashion. The same process is repeated to the columns so that, the image is enhanced in column fashion. This is the up sampling process used in the proposed method. The down sampling processes the exact reverse operation of the up sampling process. The down sampling and up sampling processes are repeated in the process. The results obtained for butterfly input are given in figure 4.5 and 4.6.

![Figure 4.4 Up sampling and Down sampling](image_url)
4.2.3 Constrained Optimization

In the optimization step, the artifacts (noises) in the images are identified. The flow chart is given in figure 4.7. The sampled images are divided into 8x8 patches. Optimization problems are then employed in the process. The regularized parameter is used for the identification of the noises in the images. Using constrained optimization process, the relationship between the original
pixel information and the magnification rate can be identified. The optimization problem helps in defining the artifacts in the images. The identified optimized image pixels are then combined to form the patch again. Finally, the optimized image is obtained and it is resized to the enhanced image size.

![Flowchart of constrained optimization](image)

**Figure 4.7 Flowchart of constrained optimization**

### 4.2.4 NLM Filter

The optimized images are then filtered using NLM filter. The steps involved are given in figure 4.8. The NLM filter is an extension of neighborhood filtering algorithms. It is based on the assumption i.e., that image content is likely to repeat itself within some neighborhoods and in neighboring frames. It computes denoised pixel by the weighted sum of the surrounding pixels of (within frame and in the neighboring frames). Natural images also have enough redundancy to be restored by NL-means. Non-Local means filtering takes a mean of all pixels in the image, and are weighted by how similar these pixels are to the target pixel. This results in much greater post-filtering clarity, and less loss of detail in the image compared to the local mean algorithms.
4.2.5 Structured Sparse Representation

In the Structured sparse represented process, the images filtered using NLM filter, are enhanced by the generation of the sparse matrix based on the identification of the data fidelity term, sparse regularization term and the gradient regularized term. The generated sparse representations are updated based on the identification of the sparse coding coefficients. For the updation of the sparse matrices, the convex optimization problem is solved. The steps involved are given in figure 4.9.

The generated optimization problem is minimized to a greater range. The formulated optimization problem can be defined as non-smooth non-convex objective function. The sparse term helps in the removal of zero elements in the identified image. The generated sparse terms are updated and the resulting final updated enhanced images denote the contrast enhanced super-resolution image. The sparse terms generated in each step are checked in each instant so that, the resulting images are enhanced. For checking the sparse terms generated, the soft
thresholding process is employed. When a sparse coding algorithm is applied to natural images, the learned bases resemble the receptive fields of neurons in the visual cortex. The goal of sparse coding is to represent input vectors approximately as a weighted linear combination of a small number of (unknown) “basis vectors.” These basis vectors thus capture high-level patterns in the input data. Sparse codes are a favourable compromise between dense and local codes. The small representational capacity of local codes can be remedied with a modest fraction of active units per pattern because representational capacity grows exponentially with the average activity ratio. Thus, distinct items are much less likely to interfere, when represented simultaneously. Furthermore, it is much more likely that a single layer network can learn to generate a target output, if the input has a sparse representation. This is due to the higher proportion of mappings being implementable by a linear discriminant functions. Learning in single layer networks is therefore simpler, faster and substantially more plausible in terms of biological implementation. The result of sparse coding for butterfly image is given in figure 4.10.

![Figure 4.9 Flowchart of Structured Sparse Representation](image-url)
The input butterfly image is processed with the SMSR technique for single frame super-resolution to obtain the high resolution frame. The output is given in figure 4.11. From the output, the inference has been made that the original color information are well preserved in the resulting image. The up sampling and down sampling will make the scaling of the images in different levels and also in each levels, the original pixel information are well preserved.

Figure 4.11 (a) Input butterfly image (b) Reconstructed image by SMSR
4.3 RAPID ADAPTIVE SUPER-RESOLUTION TECHNIQUE

The technique of denoising is incorporated in getting high resolution video. Except the concept of fuzzy logic, the remaining methodologies are applied. The reason for not using the fuzzy rules is that without using the fuzzy rules while executing the denoising program, the adaptive median filter and the weighted mean filter give poor result. Because, these filters never check whether the pixel is noisy or not. Thus, applying the concept of denoising in all the pixels leads to smoothing the image. So, the edges are not clear and the image is also blurry. To provide extra knowledge regarding the noise levels of the pixels, the fuzzy rules are the perfect choice. They instruct the denoising programs where to execute and not to execute. Hence, they provide good result by keeping the image features intact. As far as the super-resolution is concerned, for producing high resolution frames, no such extra knowledge is needed. Thus the fuzzy rules are not used here.

4.3.1 RAST ALGORITHM

*Step1:* The LR video sequences are given as the input

*Step2:* The frames are separated into non-overlapping blocks, the keyframes are extracted and the Motion Vectors (MV), the Residual Error (RE) are calculated between the frames and stored in an array

*Step3:* The SMSR technique is applied on the low resolution keyframe1 to obtain the corresponding high resolution keyframe1

*Step4:* The blocks in the high resolution keyframe1 is transferred to the low resolution between frame1 to obtain the corresponding high resolution frame using the COPY function

*Step5:* The blocks in the high resolution between frame1 is transferred to the low resolution between frame2 to obtain the corresponding high resolution frame using the same COPY function
Step6: The step5 is executed until the new keyframe is reached after that the step4 is executed once to get the high resolution frame of newly found keyframe

Step7: The output obtained is the high resolution video sequences

Thus the low resolution video sequences are converted into high resolution video sequences within the short period of time. The pictorial representation of the algorithm is given in the Fig. 4.13.

The motion vectors are calculated by using the simple block matching algorithm of average absolute differences. There are more searching algorithms available like Exhaustive search, Three step search, New three step search, Simple and efficient search, Four step search, Diamond search, Adaptive road pattern search, etc., The Residual Error value is calculate image differencing method.

If the super-resolution technique is executed with any of the video coding standards, the block number, motion vector and the residual error value is taken from the stored array values. It will be of great use while applying the super-resolution technique to generate high resolution frames.
Figure 4.12 Block diagram of RAST
The foreman low resolution (32x32) image is taken as input and it is separated into blocks. It is given in the figure 4.12. Then, the Motion Vectors (MV) and the Residual Error (RE) are calculated.

Figure 4.13 The LR foreman frame1 is separated into non overlapping blocks

The first two low resolution frames are taken. The pixel differences between them are calculated using the pixel difference. The two low resolution images, the keyframe1 and the between frame1 are denoted as $K^l_1$ and $B^l_1$. For convenience, these two frames are denoted in terms of $previous^l_{i-1}$ and $current^l_i$ frame, respectively. That is, $K^l_1$ is considered as $previous^l_{i-1}$ frame and $B^l_1$ as $current^l_i$. Here, $l$ in the super script represents the low resolution frame. The pixel difference is calculated as,

$$D(m,n) = \left| previous^l_{i-1}(m,n)^l - current^l_i(m,n)^l \right|$$ (4.1)

Thus, the absolute difference between the two frames is calculated for all the blocks and the error frame $I_d(i,j)$ is obtained. These error values are denoted as Residual Error (RE). The residual error without motion compensation of two low resolution images is given in the figure 4.14. Then the motion is
estimated using the Average absolute differences technique. The formula for this AAD is that

\[
AAD = \frac{1}{x^2} \sum_{m=1}^{x} \sum_{n=1}^{x} |x_{m,n}^{l,i} - x_{m,n}^{l,i}| 
\]

(4.2)

\[
(4.3)
\]

Figure 4.14 (a) foreman original frame1 (b) frame2 (c) Residual Error without motion compensation

The motion vector is obtained and it minimizes the value. The low resolution keyframe1 is given as an input to any of the single frame super-resolution technique. In this research work, the SMSR technique is applied to get the high resolution of previous\(^h\)\(_{t-1}\). The resultant high resolution frame is previous\(^h\)\(_{t-1}\). It is written as,

\[
previous_{t-1}^h = \text{SMSR}(previous_{t-1}^l) \tag{4.3}
\]

The high resolution keyframe1 obtained is applied to get the high resolution frame of between frame1. This method is achieved by using the copy function.

\[
current_{t}^h(x) = \text{copy}(previous_{t-1}^h, current_{t}^l) \tag{4.4}
\]

Here \(current_{t}^h(x)\) is predicted from previous\(^h\)\(_{t-1}\). During the copy function, the motion vector \(mv_{t-1,1}(x)\) and the residual error should be added. If
we add the residual error without estimating the motion vector, it smoothes the output. In the figure 4.15, it is realized that if the error frame between the frames 1 and 2 is added with the frame1 directly, it gives the frame2. But, the result does not exactly match with the original frame2 and the comparison is given in the figure 4.16. It depicts that the original high resolution frame2 and the frame2 generated by adding the error value without motion compensation with the frame1 are different from each other. It depicts the importance of motion compensation while adding the error value.

**Figure 4.15** (a) Foreman frame1 (b) Error frame of 1 and 2 without motion estimation (c) Result of error frame added with the frame1 to produce the frame2

**Figure 4.16** (a) Original frame2 (b) Frame2 produced by adding the frame1 and the error frame without motion estimation
The motion compensation between the foreman frame 1 and 2 is given in the figure 4.17.

![Frame 1 and Frame 2 with motion compensation](image)

**Figure 4.17 Motion compensation between two frames 1 and 2**

The motion vector is the displacement between the macro blocks in the current frame and the best matching macro block is the reference frame.

The two most popular measures to determine the best match between two blocks are defined as,

\[
MSE(X, Y) = \frac{1}{n^2} \sum_{m=0}^{n-1} \sum_{n=0}^{n-1} (X(m, n) - Y(m, n))^2
\]  

(4.5)

\[
SAD(X, Y) = \sum_{i=0}^{n-1} \sum_{j=0}^{n-1} |X(i, j) - Y(i, j)|
\]  

(4.6)

Here, X and Y are two frames, MSE is the Mean Squared Error, SAD is the Sum of Absolute Differences, (i,j) is the pixel value in the same spatial location of two frames and n x n is the size of the image. Due to the computational complexity, SAD measure is selected to predict the best match.

The original foreman frame 1 is reconstructed with SMSR algorithm and the two frames are given in the above figure 4.18. The PSNR value achieved is 31.97. This high resolution frame is given as an input to the reconstruction of
subsequent low resolution frames. Thus, the high resolution frames will be generated by executing the copy function.

Figure 4.18 (a) Original foreman frame1 (352 x 288) (b) reconstructed frame1 (676 x 598)

The flow chart of the copy function is given in the figure 4.19. The input high resolution frame is initially the first keyframe. The diagram depicts the steps involved in the Copy function. After computing the motion vector and the residual error, the blocks in the previous high resolution frame are added with the motion vector and the residual error, if motion and error are present, otherwise, just the blocks of data are copied to the current low resolution frame. This kind of copying avoids the execution of single frame super-resolution algorithm in all the individual low resolution frames. Also, the overall execution time is reduced. As the RAST technique is adaptable to any kind of single frame super-resolution technique, it is named as Rapid Adaptive Super-resolution Technique.
4.3.2 ALGORITHM FOR COPY FUNCTION IN RAST

**Step 1:** During the copy function, the input given is the high resolution keyframe \( h_{t-1} \) and the low resolution between frame \( h_t \) and \( l_t \).

\[
\text{previous}^h_{t-1} = \text{SMSR}(\text{previous}^l_{t-1})
\]  
(4.7)

**Step 2:** The two frames are split into non-overlapping blocks.

**Step 3:** For each block in the previous \( h_{t-1} \) frame,

**Step 3.1:** Presence of motion is analyzed with the current \( l_t \) frame blocks. If the motion is present, then the displacement (or motion vector \( mv \)) is added with the block’s coordinates of the previous \( h_t \) frame using the Equation (4.8)

\[
current^h_t(\alpha x) = \text{previous}^h_{t-1}(\alpha(x + mv_{t-1,t}(x)))
\]  
(4.8)

Here \( \alpha \) is the magnification factor.

**Step 3.2:** Presence of residual error is checked with the current \( l_t \) frame. If yes, then the error value is applied with the bicubic interpolation technique i.e., \( \text{BI}(RE^l_{t-1,t}(x)) \)

**Step 4:** The block values are copied to the current \( l_t \) frame. This is written in Equation (4.9) as,

\[
current^h_t(\alpha x) = \text{previous}^h_{t-1}(\alpha(x + mv_{t-1,t}(x))) + \text{BI}(RE^l_{t-1,t}(x))
\]  
(4.9)

**Step 5:** Step 3 is repeated for all the blocks in the previous \( h_{t-1} \) frame

**Step 6:** The output is the high resolution between frame \( h_t \) and \( l_t \) frame

**Step 7:** The frames are interchanged to process the video until new keyframe is reached. This is written in equation as,

\[
\text{previous}^h_{t-1} = \text{current}^h_t
\]  
(4.10)

\[
\text{current}^l_t = \text{next}^l_{t+1}
\]  
(4.11)
Step8: The steps 1 to 7 is repeated until end of the video. The output is the high resolution video sequences.

Thus, the copy function is performed to obtain the high resolution video. This function reduces the overall execution time of the processing compared to the other state-of-the-art algorithms. The algorithm is depicted in pictorial manner in the Fig. 4.19. The input high resolution frame is initially the first keyframe. The diagram depicts the steps involved in the Copy function. After computing the motion vector and the residual error, the blocks in the previous high resolution frame are added with the motion vector and the residual error, if motion and error are present, otherwise, just the blocks of data are copied to the current low resolution frame. This kind of copying avoids the execution of single frame super-resolution algorithm in all the individual low resolution frames. Also, the overall execution time is reduced.
Figure 4.19 Flowchart of copy function in RAST
The zooming can be performed for various levels which the user is interested to view. The zooming factor is denoted as $\alpha$. The previous high resolution frame is written as $\text{previous}_{i-1}^h(\alpha x)$. According to the zooming factor, while copying the high resolution frame to low resolution frame, the magnification should be done in the error value calculated too. To do this, a basic interpolation technique is applied. In this research, bicubic interpolation technique is utilized to magnify the error value. Here, the concept is to magnify the video. While magnifying the previous frame according to the factor $\alpha$, the next frames are also viewed in magnified version. To achieve such magnification, while adding the error with motion compensation, the corresponding error should be magnified because, the error value is calculated between the original low resolution frames. So, performing bicubic interpolation in the error value compensates such magnification.

\[
\text{current}_{i}^h(\alpha x) = \text{previous}_{i-1}^h(\alpha(x + \text{mv}_{i-1,i}(x))) + BI(RE_{i-1,i}^l(x)) \tag{4.12}
\]

Here, $\text{current}_{i}^h(\alpha x)$ gives the magnified version of high resolution frame to be obtained from $\text{previous}_{i-1}^h(\alpha x)$ for the scaling factor $\alpha$. The value $BI(RE_{i-1,i}^l(x))$ gives the bicubic interpolation of the error value. The nature of bicubic interpolation is that it interpolates the empty location by getting the information from an original pixel and sixteen of the surrounding pixels. These details are used to determine the color to fill the empty locations (new pixels) that are created from the original pixel. As a result of the bicubic interpolation, the input image size is increased as twice as original in both row wise and column wise.

The figure 4.20 shows the exact technique of how the super-resolution frame is generated from the Low Resolution (LR) frames.
Figure 4.20 Example of RAST

\[ \text{Motion vector} \]

\[ \text{LR1} \quad \text{LR2} \]

\[ \text{HR1} \quad \text{HR2} \]
The low resolution foreman frames 1 and 2 are considered and the motion vector and the residual error are calculated. Then, the foreman frame 1 is given to the SMSR algorithm to get the High Resolution (HR) frame 1. The residual error is interpolated by bicubic interpolation method. The displacement and the interpolated residual error are added with the high resolution frame 1 block and the result is stored in the high resolution frame 2. Thus, the high resolution frames are generated. The final high resolution frame generated is given in the figure 4.21.

![Blocking artifact](image1.png)  ![No artifact](image2.png)

**Figure 4.21** HR image of foreman image 2 (a) generated by RAST (b) original image

The problem to be noted here is that if a block in the previous low resolution frame contains sharp edges, then the edge is preserved while running the SMSR single frame super-resolution technique on it. Consider the scenario that the residual error block also contains the sharp edge at the same location. If the bicubic interpolation is applied on the residual patch $BI(RE_{t-1,t}^{l}(x))$, the sharp edge gets blurred. While adding the residual patch $BI(RE_{t-1,t}^{l}(x))$ with the block in
the high resolution frame previous\textsuperscript{h}{i-1}(\alpha x) to obtain the high resolution current\textsuperscript{h}{i}(\alpha x) frame, at that time the ringing artifacts are introduced.

To avoid such ringing artifacts, the average absolute magnitude of the residual block is calculated. Then, a proper threshold is to be selected to decide whether to perform the copy function or not. If the edge is present in the residual, then the value calculated as the average absolute magnitude of the residual block will be high. If it is high, then the interpolation is avoided and the corresponding block in the low resolution current\textsuperscript{i} frame is up sampled using the bicubic interpolation technique BI(current\textsuperscript{i}(x)). Otherwise, the current\textsuperscript{h}{i}(\alpha x) high resolution frame is obtained by adding the previous\textsuperscript{h}{i-1}(\alpha x) with BI(RE\textsuperscript{i-1,1}(x)).

The proper threshold value is to be selected to produce the high resolution current\textsuperscript{h}{i}(\alpha x) frame. After performing extensive experiments, the threshold T value is selected as 10.

Further more, the motion vector is analyzed to check whether to perform the copy function or not, or whether to perform the bicubic interpolation of the residual or not. Consider the scenario, If the blocks in the previous\textsuperscript{h}{i-1}(\alpha x) frame contain no motion with the current\textsuperscript{i} frame, then there is no need for bicubic interpolation BI(RE\textsuperscript{i-1,1}(x)). Straightforwardly copy the contents without bicubic interpolation. It is given in the Equation (4.13)

\[
\text{current}_i^h(\alpha x) \approx \text{previous}_{i-1}^l(\alpha (x + mv_{i-1,1}(x))) \quad (4.13)
\]

Consider another scenario, If the residual error block is empty i.e., zero, then the bicubic interpolation BI(RE\textsuperscript{i-1,1}(x)) of residual is left out. Thus, the time complexity is reduced more and more and it leads to fast execution.
After completing the super-resolution of $current^k_x(\alpha x)$, the frames are interchanged as

$$\text{previous}^h_{i-1}(\alpha x) = current^k_i(\alpha x)$$  \hspace{1cm} (4.14)

$$current^l_i(x) = \text{next}^l_{i+1}(x)$$  \hspace{1cm} (4.15)

This process is executed until the next keyframe is found. The keyframes are selected by converting the low resolution frames into sequence of gray images. Then, the average absolute difference value is calculated.

$$AAD = \frac{1}{gh} \sum_{m=1}^{g} \sum_{n=1}^{h} \left| \text{previous}^l_{i-1}(m,n) - current^l_i(m,n) \right|$$  \hspace{1cm} (4.16)

The threshold value is selected to distinguish whether the current frame is keyframe or not. The threshold value based on extensive experiments is found that $Th_A$ value is 14. That is, if the threshold value is $\geq 14$, it is declared that the current frame is the keyframe, otherwise, the current frame is declared as between frame.

### 4.4 DEBLOCKING FILTER

In this research, the frames are separated as non-overlapping blocks and the frames are processed through the blocks. The advantage of using this blocking concept is that each pixel is covered by only one block. If overlapping blocks are used, the pixels in the frame are covered by multiple blocks. It leads to more computational complexity. But, the disadvantage of the non-overlapping blocking concept is that, it creates fake edges on the block boundary. A little difference on both the sides of the block boundary creates artificial edges. To resolve this tie up, a deblocking filter is applied. This filter preserves the original edges and removes fake edges created artificially. Comparing the time complexity of using the overlapping blocks, and the execution of deblocking
filter, the time complexity of the deblocking filter is less in the non overlapping blocking concept.

4.5 SUMMARY

This chapter discusses the concepts involved in the proposed RAST system to produce high resolution frames from the low resolution frames. The Structure modulated super-resolution method is implemented to produce the high resolution keyframes. The modules of implementing the SMSR method are discussed. By applying the SMSR method in the low resolution keyframes, the high resolution keyframes are obtained. Then, the keyframes are copied to the between frames till the end of the video.