A Novel Video Super-Resolution Enhancement Method Based on Residual Learning Using Hidden Markov Random Fields and a New Deep Learning Network Architecture

Mahnaz Mahdizadeh[1], Ali Akbar Khazaei[1•], Seyyed Javad Seyyed Mahdavi Chabok[1] and Farzan Khatib[1]

***Abstract* -- In today's world, improving the quality and clarity of videos has become increasingly important, particularly in the fields of surveillance, medicine, and imaging technologies. Traditional super-resolution methods primarily focus on the full reconstruction of video frames, which poses challenges in preserving fine details and complex structures. This paper introduces a novel approach based on parallel deep networks, effectively enhancing video quality by dividing video frames into three separate input branches: raw images, outputs based on Hidden Markov Random Fields (HMRF), and temporal images. The method also leverages techniques such as residual learning and random patching within a unified framework that combines spatial segmentation (HMRF) and temporal information. This integration allows the model to better capture spatial and temporal dependencies, leading to more accurate and efficient video frame reconstruction. To better focus on high-frequency details and mitigate the vanishing gradient problem, residual learning is employed, enabling the network to estimate only the additional details necessary for reconstructing high-resolution images. Additionally, through random patching, the network training process is designed to emphasize critical features and intricate textures. Experimental results demonstrate that the proposed method achieves an SSIM of 0.92857 and a PSNR of 34.8617, offering superior clarity in video reconstruction.**

***Index Terms*-- Super-resolution, deep learning, Hidden Markov Random Fields, residual learning, random patching**

I. INTRODUCTION

The demand for high-quality video content is increasing across various fields, but transmitting high-resolution videos requires significant bandwidth and storage space. Therefore, video compression has been proposed as a solution to manage this issue, but high-quality video reconstruction after compression remains challenging. Enhancing the resolution of low-quality videos using super-resolution techniques has become an important research area, facing challenges such as balancing resolution and reducing artifacts in dynamic scenes.

In real-world video super-resolution, significant challenges exist, including the diversity and complexity of degradations, which affect both inference and training processes. Using long-range propagation in cases of mild degradations may improve performance, but in the presence of severe degradations, it can lead to output quality deterioration. Therefore, an image pre-cleaning step before propagation is essential to reduce noise. Using the cleaning module designed in [1] improves the quality and efficiency of the RealBasicVSR model compared to previous methods. To train models in real-world conditions, there is a need to increase data size and computational load. A random degradation scheme, which reduces training time by up to 40% and uses longer sequences instead of larger batches to more effectively exploit temporal information, has been proposed.

Video super-resolution models are generally trained on synthetic data. Reference [2] introduces the RealVSR dataset, which includes real LR-HR videos and improves the quality of detail recovery. The RefVSR method, presented in [3], uses reference videos for higher accuracy. Research in papers [4] and [5] emphasizes the challenges of recurrent models and video compression. Paper [6] addresses the role of matching in transformers for VSR, suggesting that patch matching, instead of pixel matching, leads to better performance. Temporal modeling in video super-resolution is of great importance. Some methods use optical flow or convolution to compensate for motion, which may add complexity to the model and lead to issues in certain conditions. The study [7] suggests calculating temporal differences between frames and dividing pixels into two subsets. Experiments show that this method performs well compared to others. Study [8] introduces a new transformer for compressing video super-resolution, performing self-attention in the space-time-frequency domain, and its results significantly outperform other methods. New methods for small video super-resolution using the Knowledge Transfer (STD) approach improve performance in resource-constrained conditions [9]. VISCA, an edge-assisted video streaming solution, combines super-resolution and storage, significantly improving video quality compared to existing solutions [10]. Paper [11] presents a simple and effective method for super-resolving high-quality videos from low-resolution ones, resistant to large motions and offering better generalizability. Spatiotemporal deep neural networks have shown promising results in video super-resolution (VSR) in recent years. Paper [12] introduces a new Spatio-Temporal Matching Network (STMN) that reduces dependency on motion estimation by working in the wavelet domain. This architecture includes three main components that help extract spatial and frequency information and reconstruct high-resolution frames. Paper [13] introduces a

1. Department of Electrical Engineering, Mashhad Branch, Islamic Azad University, Mashhad, Iran.

• Corresponding author Email: khazaei@mshdiau.ac.ir

Cyclic Mutual Learning Network (CycMuNet) that leverages mutual learning between spatial and temporal video super-resolution and aids in high-quality video reconstruction. Furthermore, paper [14] presents a deep learning-based SR method called FOCAS that reduces computational load and decreases delay by 50-70%. Paper [15] introduces a lightweight network called ELNVSR, which extracts spatial information using multi-group blocks and performs well while maintaining a small number of parameters. Finally, paper [16] proposes a Matching Flow Estimation (MFE) module that improves alignment performance in variable conditions by predicting coarse positions.

Super-resolution (SR) involves generating high-resolution (HR) video frames from low-resolution (LR) frames. Paper [17] introduces the MP3D network, which uses 3D convolutions to capture temporal correlations in LR frames and reduces the need for motion compensation. This network includes pyramid sub-networks, SR reconstruction, and detail refinement to enhance HR frame quality. Paper [18] presents a novel video super-resolution (VSR) method using convolutional neural networks (CNNs), which uses spatial and temporal information to improve reconstruction quality and reduce training time. It also introduces a motion blur compensation scheme. Paper [19] introduces a method to enhance the quality of encoded HEVC frames using a multi-frame in-loop filter (MIF), improving the quality of each encoded frame by utilizing spatial and temporal information from higher-quality frames. Reference [20] discusses the application of deep learning in video compression, introducing several new tools to enhance coding efficiency. Reference [21] presents an innovative video compression model that improves video frame quality at low bitrates by using adaptive sampling and block patching, significantly enhancing video quality compared to HEVC and other methods.

Overall, previous methods have made significant progress in the field of super-resolution and video file quality enhancement. However, the proposed methods have not simultaneously addressed various key features of a video file, including edge and key region detection, frame segmentation based on brightness change features, and dynamic and static object features in frames to improve performance. Additionally, the lack of new deep learning techniques, such as residual learning and random patching, can lead to the reduced performance of a powerful method. To address these challenges, this paper proposes a new method for improving video super-resolution.

This paper proposes an innovative method for enhancing video frame quality by combining multiple image inputs and a parallel network architecture. This method focuses particularly on residual image estimation, allowing the network to effectively model complex, high-frequency details. The proposed method effectively identifies spatial and temporal dependencies in video frames by utilizing advanced techniques such as HMRF-EM for segmentation and integrating temporal inputs. Although the approach builds on existing methodologies, such as residual learning and random patching, the novelty lies in integrating these techniques within a unified framework that combines spatial segmentation and temporal information. This integration enables the model to better capture both local and global dependencies in video sequences, leading to improved performance over traditional methods. This research is especially significant in the fields of video processing, image quality enhancement, and deep learning, and it can provide new solutions for various applications in these areas.

II. BASIC CONCEPTS

This section presents the foundational concepts necessary to better understand the proposed method.

A.Color Spaces in Image Processing

The use of color spaces in image processing holds significant importance as color information aids in identifying key features of an image [22]. Below, two major color spaces are introduced**:**

• Understanding the RGB Color Space

The RGB color space is one of the most widely recognized color spaces in image processing. In this space, each pixel is composed of a combination of varying intensity values of three color components: Red (R), Green (G), and Blue (B). The intensity values of these components range between 0 and 255. For instance, (255, 0, 0) represents red at maximum intensity, (0, 255, 0) represents green, and (0, 0, 255) represents blue. Due to its simplicity and compatibility with most systems, this color space is extensively used for image display and analysis [23].

• Understanding the YCbCr Color Space

The YCbCr color space consists of three main components: the luminance component (*Y*) and two chrominance components (*Cb* and *Cr*), which contain color information. The *Y* component represents brightness levels and is considered the grayscale component of the image. The *Cb* and *Cr* components indicate the color differences along the blue and red axes, respectively. This color space is especially useful in applications requiring the separation of brightness and color, such as image compression and image transmission in television networks [24].

• Color Space Conversion

Color space conversion is a critical step in image processing that enables the transformation of images from one color space to another. These conversions are performed for purposes such as color correction, noise removal, or image compression. For example, converting from RGB to YCbCr allows the separation of color and brightness information, enabling the utilization of each component's features for various processing tasks [25]. The conversion from the RGB space to YCbCr can be achieved using the following equations:

$$Y = 0.299 * R + 0.587 * G + 0.114 * B \quad (1)$$

$$Cb = -0.1687 * R - 0.3313 * G + 0.5 * B + 128 \quad (2)$$

$$Cr = 0.5 * R - 0.4187 * G - 0.0813 * B + 128 \quad (3)$$

In these equations, ***Y*** represents the luminance (brightness) component, while ***Cb*** and ***Cr*** denote chrominance (color difference) components [26].

B:Convolutional Neural Networks (CNNs)

Convolutional Neural Networks (CNNs) are a type of deep neural network specifically designed for image processing and related tasks. These networks comprise various layers, including convolutional layers, pooling layers, and fully connected layers, which collectively extract important features from images.

In each convolutional layer, information from the input image is processed using small filter windows (kernels) through convolution operations, extracting various features. Pooling layers then reduce the dimensionality and emphasize more significant features. Finally, fully connected layers transform the extracted features into one or multiple outputs for specific tasks such as pattern recognition, image classification, or high-resolution image reconstruction [27].

The CNN architecture includes several convolutional and pooling layers that sequentially extract image features and reduce their dimensions. Padding is used to control the output image dimensions, ensuring that the output has the same size as the input image [28].

CNNs are applied to high-resolution imaging and the enhancement of low-resolution images from two perspectives. For high-resolution imaging, CNNs increase image clarity using information from lower-resolution images. Through deep learning algorithms and features extracted from higher-resolution images, these networks can reconstruct high-resolution images, which have applications in fields like medicine and engineering.

For low-resolution image enhancement, CNNs serve as an effective tool for correcting and improving images. These networks significantly enhance the quality and resolution of low-resolution images, with wide-ranging applications in areas such as digital imaging, films, and medical imaging [29, 30].

C: Canny Edge Detector

The Canny edge detector is one of the most reliable and popular methods for edge detection in images, based on identifying sudden changes in color or light intensity. The operation of this detector includes several key steps:

- **Noise Removal**: Using a Gaussian filter to eliminate noise in the image and smooth it.
- **Gradient Calculation**: Determining the direction and magnitude of intensity changes by computing the image gradient.
- **Edge Enhancement**: Applying Sobel filters to extract edges and align them with the detected edges.
- **Thresholding**: Identifying significant edges and discarding unnecessary ones by applying a threshold to the image gradient.

Due to its high precision, adaptability, and simplicity, the Canny detector is among the most widely used edge detection methods in image processing [31, 32].

D: Hidden Markov Random Fields

Markov Random Fields (MRFs) are an important and practical approach in image processing for modeling spatial relationships among different pixels in an image. These fields are defined as a data structure of random variables, each dependent on the previous ones. In other words, the information of each pixel is modeled by considering its contextual and neighboring data. The advantages of these fields include improved accuracy and quality in image reconstruction and enhancement. MRF algorithms can effectively reduce noise and improve the performance of image-processing tasks [33, 34].

• Applications of Markov Random Fields in Image Reconstruction and Enhancement

Markov Random Fields are widely used for reconstructing and enhancing images. They can precisely restore noisy or low-quality images using neighboring information. Moreover, they can correct imperfections and noise in images, providing more realistic image quality. These fields are also effective in enhancing images by adjusting colors, textures, and structures, thus aiding in improving image quality and reconstruction [35].

• Structure of the Hidden Markov Random Fields Algorithm

The theory of Markov Random Fields is a subset of probability theory that examines how spatial or contextual connections among physical phenomena affect each other. MRFs are widely applied in computer vision tasks, such as image segmentation and surface reconstruction.

The Hidden Markov Random Field (HMRF) model represents a stochastic process generated by an MRF. In this model, the sequence of states cannot be directly observed but can be inferred indirectly through observations. This model is effectively used for segmenting static image regions, particularly in applications involving brain MRI images. When an image is represented by $y = (y_1, \ldots, y_N)$ with each y_i corresponding to the intensity of a pixel, the objective is to determine a set of $x = (x_1, \ldots, x_N)$, where each x_i belongs to a set L containing all possible labels. For instance, in a binary segmentation scenario, L includes {0, 1}. Using the Maximum A Posteriori (MAP) estimate criterion, the goal is to determine the optimal label x^* that satisfies the condition.

$$x^* = \underset{x}{argmax}\{P(y|x,\Theta)P(x)\} \quad (4)$$

The prior probability, $P(x)$, follows the Gibbs distribution, while the joint probability of the occurrence of the conjugate probability is expressed as shown in Equation (5):

$$P(y|x,\Theta) = \prod_i P(y_i|x,\Theta) = \prod_i P(y_i|x_i,\theta_{xi}) \quad (5)$$

The probability $P(y_i|\boldsymbol{x},\Theta)$ follows a Gaussian distribution with parameters $\theta_{xi} = (\mu_{xi}, \sigma_{xi})$. The parameter set $\Theta = \{\theta_l \mid l \in L\}$ is estimated using the EM algorithm [35].

III. METHODOLOGY

The first step in the proposed method is extracting frames from the video, where each frame *f* contains pixel information of a scene at a specific moment. These frames are in the form of color images with three RGB channels, and to enhance clarity, all channels must be improved. However, studies have shown that the human visual system is more sensitive to brightness variations than to color changes. Thus, the super-resolution process can be optimized by focusing on improving the brightness component.

By converting images from the RGB color space to YCbCr, brightness and color information can be separated. In this new space, brightness (*Y*) is stored in a separate channel, while color information is contained in two channels (*Cb* and *Cr*). This transformation allows processing to focus on the *Y* channel for frame quality enhancement, leaving the *Cb* and *Cr* channels unchanged. In the end, these are combined with the enhanced *Y* channel to produce the final improved image. This method optimizes computational resources by prioritizing brightness enhancement.

After initial frame processing and conversion to single-channel images, the next step is determining the inputs required for clarity enhancement. The first and primary input consists of raw, low-quality images. Since the goal is to improve the quality of these images, the desired output should have a similar overall structure, differing only in details. Therefore, low-quality images carry significant information about the desired output and are used as the primary input.

Regarding other input information, the first step involves applying certain transformations to better understand the general state and objects within the image. The first transformation includes segmenting video frames using the HMRF-EM algorithm (Hidden Markov Random Field with Expectation-Maximization). This algorithm serves as a powerful tool for image segmentation by dividing the image into meaningful regions based on visual feature similarities. A key advantage of HMRF-EM is its ability to model spatial relationships between neighboring pixels and statistical image features, making it suitable for analyzing complex image structures and extracting diverse content.

Using the HMRF-EM algorithm for image segmentation offers several benefits in improving super-resolution performance. One such advantage is maintaining structural consistency; as in super-resolution, preserving the structural stability of frames is crucial for generating visually pleasant and coherent results. By incorporating spatial dependencies through the HMRF model, the algorithm ensures that segmentation decisions in neighboring frames are consistent and appropriate, leading to smoother transitions and more natural clarity enhancement. Additionally, video frames often include complex structures like textures, patterns, and objects of various shapes and sizes. The HMRF-EM algorithm's ability to capture both local and global meaningful information allows it to effectively segment images with diverse and complex structures, ensuring accurate delineation of the desired regions. Finally, this algorithm provides flexibility in modeling the statistical features of intensity variation within segments, making it adaptable to various types of images and complexity levels, which makes it suitable for a wide range of applications, including super-resolution.

The segmentation process using HMRF-EM involves the iterative estimation of model parameters and updating pixel segmentation labels based on these estimates. The Expectation-Maximization (EM) algorithm facilitates this iterative process by alternately computing the expected value of hidden variables and maximizing the likelihood of observed data. In the context of super-resolution, the segmentation outputs from HMRF-EM provide valuable information about the spatial distribution of image details and structures. In this study, two outputs—binary and multi-class—are used as extracted features for further processing. The binary output highlights important regions of the image, while the multi-class output offers a more detailed representation of the image content, enabling targeted enhancement of segmented areas and focusing on specific details.

To preserve image clarity and prevent blurring, the edges of each frame *f* are identified using the Canny edge detection method, which detects local maxima in the image intensity gradient. This method calculates the gradient using a Gaussian derivative filter, highlighting abrupt brightness changes. Then, with two thresholds, strong and weak edges are identified, providing greater accuracy in the presence of noise and in detecting weak edges. After segmenting video frames with HMRF-EM and extracting edges with Canny, this information is combined to create three-channel images used as inputs for subsequent stages. These inputs provide rich information about the spatial distribution of image features and edges. Fig. 1 illustrates an example of these three features for an image.

Fig. 1. Example of 3 images forming a 3-channel input based on HMRF

In this paper, in addition to the HMRF-based inputs, temporal inputs are also proposed to incorporate temporal information as input for the quality enhancement process. These inputs are constructed by combining three consecutive low-resolution raw frames, including the previous frame f_{t-1}, the current frame f_t, and the next frame f_{t+1}. For the first and last frames, where there is no preceding or succeeding frame (there is no f_{-1} or $f_{t'+1}$ where t' is the last time sample), the first and last frames are repeated to provide temporal inputs for all frames and capture the temporal dependencies between neighboring frames. In other words, the first frame sequence is $\{f_0, f_0, f_1\}$ and the last frame sequence is $\{f_{t'-1}, f_{t'}, f_{t'}\}$.

Integrating temporal inputs identifies the static and dynamic components in the video, which can assist in improving super-resolution. This is because the resolution enhancement approach differs for dynamic and static objects. Dynamic objects, especially those moving at high speeds, may exhibit more motion blur compared to static objects. The proposed method's three-channel temporal inputs allow the enhancement strategy to be adjusted based on temporal characteristics, resulting in more effective resolution improvement across the entire video. Fig. 2 illustrates an example of combining three frames, clearly showing the movement of the child's head relative to other objects in the image.

With the raw single-channel images, HMRF-based images, and temporal images prepared, the proposed method's inputs for video frame enhancement are ready. To process these inputs, a novel parallel network architecture is designed and implemented, consisting of three independent branches. Each branch includes 19 convolutional layers accompanied by ReLU activation functions. The ReLU function, in addition to introducing the ability to analyze

nonlinear conditions in the network, addresses the vanishing gradient problem in deep neural networks by not saturating for large values, resulting in better network training.

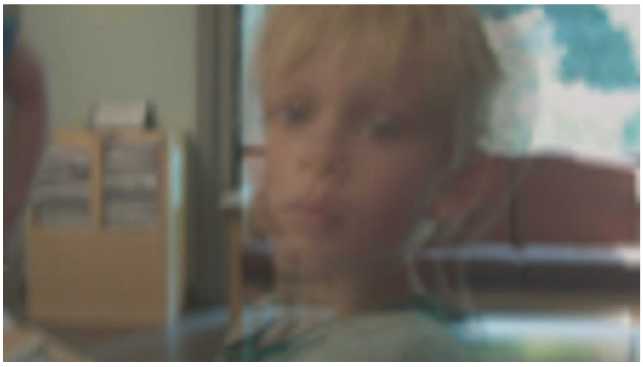

Fig. 2. Examining the dynamics and statics of objects within the video in 3 consecutive sample frames

Each branch of this parallel network is specifically designed to process its respective input type to effectively extract distinctive features associated with each input. The first branch processes the raw single-channel inputs, the second branch handles the HMRF-based images, and the third branch processes the temporal images. After each network processes its inputs through the 19 convolutional layers, the outputs of these three branches are fused into a multi-channel image using a concatenation layer. Subsequently, a final convolutional layer is applied to process and combine the information from the three inputs, ultimately producing enhanced video frames. Fig. 3 illustrates the final structure of this network.

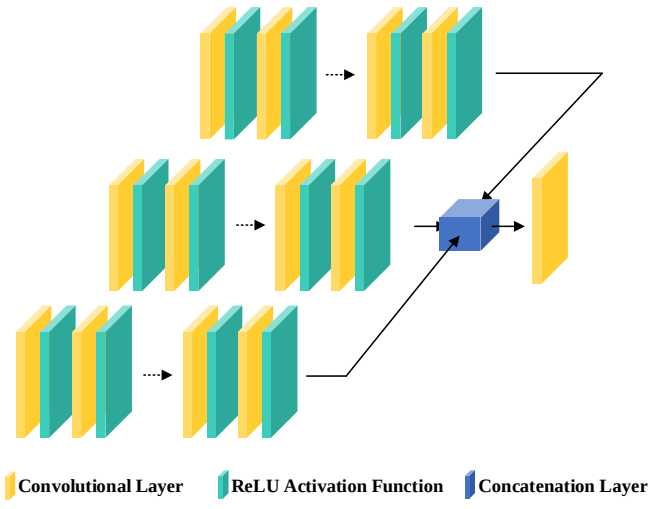

Fig. 3 . Designed parallel network structure

Adopting a parallel network approach instead of using a single sequential network to process all inputs as a multi-channel image allows the network to learn the features of each input type separately. Raw inputs, HMRF-based images, and temporal images each capture complementary and unique aspects of video frames. Raw inputs provide the overall structure of the image, HMRF-based images offer detailed edge and segmentation information, and temporal images add the dynamics and static features of objects to the network. This separation enables the network to independently learn filters and weights tailored to each input type, effectively leveraging the specific features of each input to enhance image quality.

After preparing the inputs and designing the network structure, it is essential to determine the network's output and training method. The simplest approach involves training the network directly with high-quality images as the output. However, it has been shown that this method may result in a lack of focus on details and reduced accuracy [36]. When estimating high-quality images directly, the network must predict both the details and the general structure of the image simultaneously. Since raw inputs already contain the general structure, the network's error decreases quickly during training, leading to convergence to a local optimum focused on structural estimation. As the error diminishes and the input and output become relatively similar, the network may suffer from gradient reduction, slow improvement, and suboptimal performance in estimating details.

To address this challenge, a novel training approach is proposed in this paper. Inspired by the VDSR method [36], residual learning is employed for training the network. Residual learning involves using the residual image as the target variable during training. The residual image represents the difference between the high-resolution image and its low-resolution counterpart, resized using bicubic interpolation to match the dimensions of the high-resolution image. In other words, the network learns only the details that need to be added to the low-resolution image to make it resemble the high-resolution reference. This method is effective because the network focuses on learning the residual details rather than reconstructing the entire high-resolution image, capturing textures and complex details more accurately.

Estimating the residual image offers several significant advantages that enhance performance. First, it emphasizes high-frequency details, such as textures and fine structures, allowing the network to allocate its capacity to model these complex details. Second, breaking the problem into two steps (estimating the residual image and adding it to the interpolated input) simplifies the learning process. This approach also helps convolutional neural networks to better learn the residual mapping while preserving the overall content of the input image, preventing smoothing or blurring in image regions. Consequently, the final images exhibit sharper edges and more realistic details.

It is worth noting that residual learning requires focusing on the details of each frame f rather than its general features. Using the complete Y-channel of each frame f (Y_f) may lead the network to concentrate on unnecessary details related to image content, potentially reducing accuracy in resolution enhancement. Therefore, as a preprocessing step, random patching is applied. This technique divides the input and output images into smaller, sometimes abstract, patches (e.g., part of an object's texture), enabling the network to better focus on improving critical details and resolution. Since the input and output images have different dimensions, the low-quality raw images are first resized to match the dimensions of the high-quality images using bicubic interpolation. Subsequently, other inputs (HMRF-based and temporal images) are computed using the resized images. This ensures that all images have consistent dimensions, facilitating random patching. Overall, Fig. 4 illustrates the proposed method's flowchart.

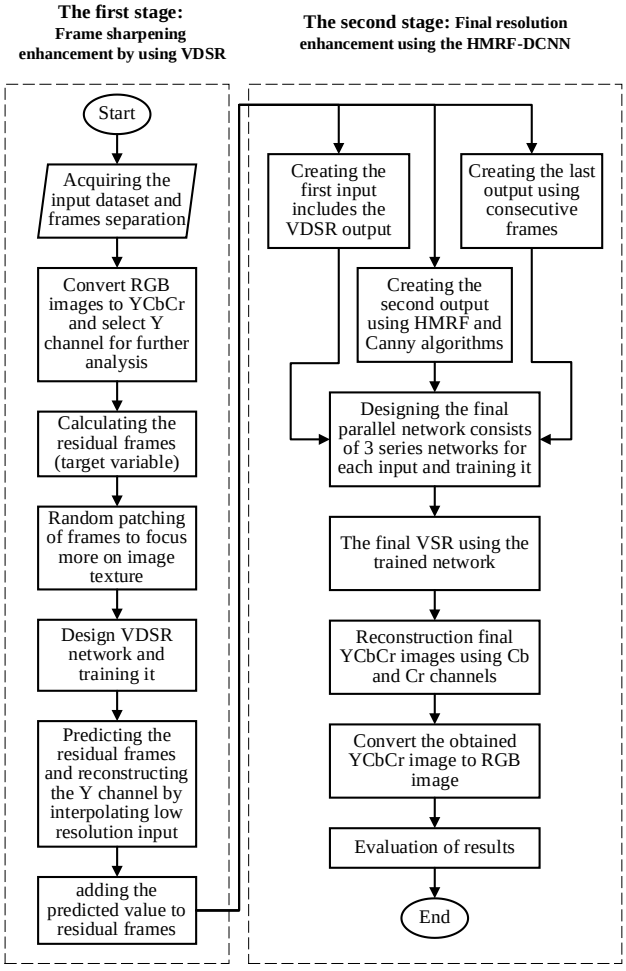

Fig. 4 . Flowchart of the proposed method

IV. DATASET

In this study, the Vimeo-90K dataset [37] is used, which is one of the largest and most comprehensive datasets for video tasks, including video super-resolution (VSR). This dataset contains 89,800 high-resolution video clips extracted from various videos such as films and personal cameras. The clips in this dataset include image sequences with complex motions and lighting changes, which pose various challenges for video processing algorithms. The Vimeo-90K dataset, due to its diversity and high quality, has been used as one of the reliable resources in research related to video resolution.

V. EVALUATION METRICS

In this section, we describe the metrics used to evaluate the quality of enhanced video frames.

• Peak Signal-to-Noise Ratio (PSNR):

PSNR measures the ratio between the maximum possible power of a signal and the destructive noise power that affects its display accuracy. This metric is defined as follows:

$$PSNR = 10.\log_{10}\left(\frac{MAX^2}{MSE}\right) \quad (6)$$

where *MAX* is the maximum pixel value of the image (255 for 8-bit images), and *MSE* is the mean squared error between the reference and the processed image. Higher *PSNR* values indicate better image quality.

• Structural Similarity Index (SSIM):

SSIM evaluates image quality based on structural information degradation, considering the brightness, contrast, and structure of images. SSIM is defined as follows:

$$SSIM(x,y) = \frac{(2\mu_x\mu_y+C_1)(2\sigma_{xy}+C_2)}{(\mu_x^2+\mu_y^2+C_1)(\sigma_x^2+\sigma_y^2+C_2)} \quad (7)$$

where $\mu_x$and μ_y are the average pixel values of images *x* and *y*, σ_x^2 and σ_y^2 are the variances, $\sigma_{xy}$is the covariance of *x* and *y*, and $C_1$and $C_2$are constant values that ensure stability in the division. The *SSIM* index values range from -1 to 1, with higher values indicating better quality.

• Mean Squared Error (MSE):

This metric measures the average squared difference between the reference and processed images and is defined as follows:

$$MSE = \frac{1}{N}\sum_{i=1}^{N}(I_{ref}(i) - I_{proc}(i))^2 \quad (8)$$

where $I_{ref}$and I_{proc} are the pixel values of the reference and processed images, respectively, and *N* is the number of pixels. Lower *MSE* values indicate better image quality.

VI. SIMULATION RESULTS

In this section, the simulation results of the proposed method are presented, and to evaluate the performance of the method, the results are analyzed. It is worth mentioning that the proposed method has been implemented on a system with the following specifications:

CPU: Core i7 13650 HX – Ram: 32Gb DDR5 – GPU: Nvidia RTX 4060 8Gb

A. Parameter Settings and Network Training

In this section, the results of the proposed method and their analysis are discussed. Initially, preprocessed video data and frames are extracted from video files. Each color video frame consists of three channels (red, green, blue), and to reduce computational load, the frames are converted to the YCbCr color space, with only the luminance channel (*Y*) being processed. Then, low-quality images are resized to 448×256 dimensions using cubic interpolation, matching the dimensions of the high-quality images. By subtracting these images, the output set is prepared, and three types of network inputs, namely the single-channel initial, HMRF-based images, and temporal images, are computed. The HMRF-based images are generated using the HMRF-EM algorithm and Canny edge detection. The HMRF-EM algorithm divides the frames into binary and multi-class outputs, capturing precise structures. For this purpose, three cases of 3, 4, and 5 classes are modeled to determine the best case. Additionally, the HMRF-EM algorithm is implemented with 10 iterations to balance processing time and accuracy. The Canny algorithm detects edges using lower and upper thresholds (0.3 and 0.75), respectively. These two thresholds provide an appropriate range for edge detection based on the diversity of the input frame images. Then, these images are merged and used as HMRF-based three-channel inputs. Finally, temporal inputs are determined from three consecutive low-resolution

frames, interpolated for the network to define the video's temporal dynamics.

Next, random patching of the images is performed to focus the network more on textures than content. Since the input images are 448×256 in size, 100 patches of size 32×32 are considered to reconstruct textures. Moreover, data augmentation is used to improve the dataset and prevent the content of the images from affecting the results. Data augmentation includes random mirroring and random rotation of up to 90 degrees for input frames. After preparing the data, the final network design is carried out. The architecture of this network consists of three parallel branches, each processing a specific type of input (single-channel initial, HMRF-based images, and temporal images). Each branch contains 19 convolutional layers with ReLU activation, and each of these layers has filters of size 3×3, focusing more on texture details, allowing the network to provide local results for texture enhancement. For each convolutional layer, 32 filters are considered, so the network has approximately 502,369 learnable parameters in total. The outputs of these three branches are merged through a final convolutional layer to produce enhanced frames. This final layer also has a filter size of 3×3, but it has only one filter that generates the remaining luminance channel image. Also, zero padding is applied to all convolutional layers to preserve the spatial dimensions of the input images.

After the design, the network is trained using the ADAM optimization algorithm with a learning rate of 0.0001, a batch size of 16, and 20 epochs. Fig. 5 illustrates the training process of this network for the HMRF 3-class case. As can be seen, the training process converged very fast showing the effectiveness of hyperparameter adjustments for network training. After training, the final enhanced frames are obtained by adding the network output and the corresponding interpolated raw frames (since the network output is the remaining frame). These are then merged with the initial Cb and Cr frames to produce the final enhanced image.

After training the network in the 3-class, 4-class, and 5-class cases, a comparison is made between these cases. The MSE metric for all frames in these three cases is calculated, and their final averages are reported in Table I. As seen, the 3-class and 4-class cases have almost similar performance, but the 4-class case provides better performance, indicating the impact of increased complexity. However, the 5-class case shows weaker performance, indicating that increasing the number of segmentation classes does not always improve performance. Finally, the 4-class case is selected as the optimal case for further processing, and evaluation metrics for each frame are calculated and analyzed.

The computational complexity of the proposed method is justified given its application in video resolution enhancement, where high-quality output is prioritized over minimal computational cost. Each of the three parallel CNN branches consists of 19 convolutional layers, leading to an increased number of operations; however, the model remains efficient due to optimized layer design and parallel processing. Moreover, although the network structure seems complicated, it has a small number of trainable parameters (502.3k parameters), less than many more straightforward structures. For example, the proposed network has fewer trainable parameters than the deep learning network mentioned in [38], which just has two fully connected layers. Additionally, preprocessing operations are implemented in an optimized manner to ensure minimal overhead. While real-time performance depends on the available hardware, the method achieves an inference speed of 33.72 fps on a GeForce RTX 4060 – 8GB GPU model, demonstrating its scalability for practical applications in video processing.

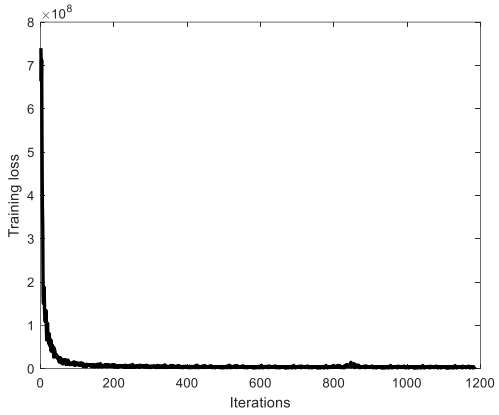

Fig. 5. The training process of the network designed for HMRF set in 3-class mode

TABLE I

Study of the Effect of Segmentation Type on the Performance of the Proposed Method

HMRF class number	3	4	5
Average MSE	37.5962	36.8434	42.3238

B.Performance Evaluation of the Proposed Method Using the PSNR Metric

As explained in the previous section, PSNR measures the ratio between the maximum possible power of a signal and the destructive noise power that affects its display quality. This metric is expressed in decibels (dB), and higher PSNR values indicate better image quality, as they reflect a lower level of distortion or noise caused by the enhancement process.

Typically, PSNR values above 30 dB are considered acceptable and suitable for most applications. In this section, the performance of the proposed video frame super-resolution enhancement approach is evaluated using the PSNR metric. In this regard, the PSNR metric is first calculated between all the enhanced images produced by the proposed method and the reference high-quality images. The average PSNR value calculated for all frames using the proposed method is 34.8617 dB, higher than the 30 dB threshold, indicating that the proposed method performs well. Furthermore, two frames from randomly selected video sequences are presented, and their results are shown in Fig.s 6 and 7. For instance, as can be seen in Fig. 6, the proposed method could greatly enhance the image's resolution. The PSNR values for these two frames are 39.0568 and 31.5714, respectively. Achieving PSNR values in a range significantly above the ideal threshold (30 dB) demonstrates the effective enhancement of the proposed method.

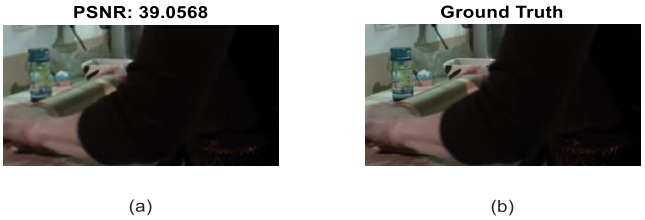

Fig. 6 - Result of improving the hyper-resolution in the first random frame sample along with the obtained PSNR value (a) Improved frame using the proposed method (b) Frame with reference quality

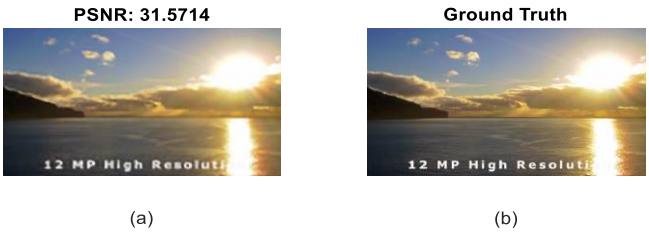

Fig. 7 - Result of improving the hyper-resolution in the second random frame sample along with the obtained PSNR value (a) Improved frame using the proposed method (b) Frame with reference quality

C: Performance Evaluation of the Proposed Method Using the SSIM Metric

In this section, we evaluate the performance of our proposed super-resolution enhancement method using the SSIM metric. As previously mentioned, SSIM values range from -1 to 1, with higher values indicating better image quality. An SSIM value close to 1 indicates that the processed image is very similar to the reference image in terms of brightness, contrast, and structure. To provide a visual representation of SSIM, the SSIM map for a randomly selected frame is shown in Fig. 8. In this map, dark areas indicate regions with low SSIM values, meaning the structural similarity between the reference and processed images is low. Conversely, bright areas represent regions with higher SSIM values, indicating greater structural similarity.

As shown in Fig. 8, the SSIM map generated by the proposed method predominantly has lighter colors, suggesting that most of the frames examined have a high structural similarity with the reference image.

Additionally, to comprehensively assess the performance of the proposed method, the SSIM values for all frames were calculated, and the average was taken. The resulting value is 0.92857, indicating the effective performance of the proposed method.

Overall, the SSIM evaluation confirms that our approach not only enhances the image clarity but also preserves the important structural details of the original frames, resulting in high-quality video output.

Fig. 8. SSIM map for a random frame sample (a) Improved frame using the proposed method (b) SSIM map of the same frame

D: Performance Evaluation of the Proposed Method Using the MSE Metric

In this section, we evaluate the performance of our proposed approach using the MSE metric. To show the distribution of MSE values across different frames, a histogram is presented in Fig. 9. The provided histogram illustrates the frequency distribution of MSE values for all frames processed by the proposed method.

As shown in Fig. 9, the highest concentration of MSE values for all frames falls within the range of 0 to 20. This indicates that most of the frames processed by our method have very low MSE values, demonstrating the effectiveness of our approach in minimizing errors and discrepancies from the reference images. Furthermore, the low spread of MSE values highlights the consistency and reliability of the proposed method in maintaining high-quality results across various frames.

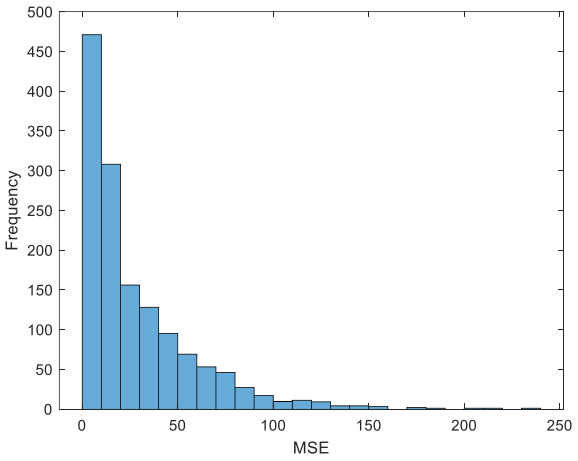

Fig. 9 - Histogram of MSE values obtained for all frames

VII. COMPARISON

The field of VSR has progressed substantially in recent years, with numerous methods developed to improve video quality and sharpness. This paper examines seven key studies and contrasts their findings with the proposed approach in this work.

Reference [39] introduces a network called EGVSR, which achieves faster processing and reduced computational load by utilizing spatiotemporal learning and hardware optimization. This method is faster and visually superior to TecoGAN. Reference [40] uses deep convolutional neural networks (CNNs) to estimate blur kernels and recover video frames, achieving better results in terms of PSNR and SSIM compared to existing methods. Additionally, reference [41] proposes a network called DDAN that uses spatiotemporal features for motion compensation and more accurate reconstruction, demonstrating superior performance in various tests. Reference [42] presents a hierarchical combined method that recovers lost details and effectively handles complex motion in videos by dividing the input sequence into specific groups and using attention and fusion modules. Reference [43] proposes a self-supervised learning framework that eliminates the need for predefined degradation models or paired HR-LR training data. Their method jointly estimates blur kernels and reconstructs high-

resolution videos while leveraging optical flow for temporal information.

In contrast, the challenge of generating high-quality, high-frame-rate videos was addressed by introducing a hybrid imaging system (HIS-VSR) in reference [44]. Their model integrates a low-resolution, high-frame-rate main video with a high-resolution, low-frame-rate auxiliary video, effectively combining super-resolution and frame interpolation techniques. This method outperforms conventional video processing models, such as Deep-SloMo, particularly in reconstructing dynamic scenes with enhanced spatial-temporal details. Paper [45] enhances a pre-trained image super-resolution network with a fast temporal aggregation module that employs deformable convolutions for inter-frame alignment.

The proposed method in this paper enhances video super-resolution by combining Hidden Markov Random Fields (HMRF) and deep learning networks. Utilizing a combination of inputs, including HMRF-based input, raw low-quality images, and temporal input, improves the resolution of frames. The proposed method enhances the super-resolution performance compared to other methods through residual learning and a new parallel network architecture. Comparisons show that the proposed method outperforms others in PSNR and SSIM metrics and demonstrates high robustness in producing high-resolution videos (Table II).

TABLE II

Comparison of the Proposed Method with Other Meta-separability Methods

Reference	Method	PSNR	SSIM
[39]	EGVSR and CNN (FNet and SRNet)	25.88	**0.80**
[40]	Deconvolution-based Blind Video Super-resolution using CNNs	29.18	**0.8372**
[41]	DDAN	26.48	**0.7892**
[42]	TGA	27.59	**0.8419**
[43]	A self-supervised learning method	24.59	**0.7629**
[44]	HIS-VSR	30.99	**0.9291**
[45]	IMDN	26.34	**0.7858**
Proposed method	HMRF + Deep learning	34.86	**0.9285**

VIII.CONCLUSION

This paper presents an innovative approach to improving the quality of video frames through super-resolution, designed based on the combination of advanced deep learning techniques and image processing. The proposed method, utilizing a new parallel network architecture as an effective tool, successfully extracts useful information from various inputs, including raw images, HMRF-based images, and temporal inputs, simultaneously.

Using the HMRF-EM algorithm, the image segmentation process contributed to maintaining structural stability and accuracy in edge detection and key region identification. This improvement in segmentation quality, especially in images with complex and diverse structures, allowed for creating more visually pleasing and realistic results. Additionally, using temporal inputs by combining three consecutive frames enabled the network to recognize temporal dependencies between frames and adjust the enhancement strategy based on dynamic and static features.

The residual learning method, considered an innovative training approach, focused on estimating high-frequency details, allowing the network to estimate only the necessary details rather than reconstructing the entire image. This strategy was particularly effective in preserving textures and intricate details, significantly enhancing the final image quality. Furthermore, the use of random patching as a preprocessing step enabled the network to focus on key, effective details, thus preventing image quality degradation.

Experimental results demonstrated that the proposed method, achieving an SSIM value of 0.92857 and a PSNR value of 34.8617, significantly outperforms existing methods. These results indicate the effectiveness of the proposed method in reconstructing high-quality video frames with precise detail.

Finally, this research could serve as a foundation for future studies in image and video quality enhancement using advanced deep-learning techniques and open new avenues for future research in this field.

VIII. REFERENCES

[1] Chan, K. C., Zhou, S., Xu, X., & Loy, C. C. (2022). Investigating tradeoffs in real-world video super-resolution. In Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition (pp. 5962-5971).

[2] Yang, X., Xiang, W., Zeng, H., & Zhang, L. (2021). Real-world video super-resolution: A benchmark dataset and a decomposition-based learning scheme. In Proceedings of the IEEE/CVF International Conference on Computer Vision (pp. 4781-4790).

[3] Lee, J., Lee, M., Cho, S., & Lee, S. (2022). Reference-based video super-resolution using multi-camera video triplets. In Proceedings of the IEEE/CVF conference on computer vision and pattern recognition (pp. 17824-17833).

[4] Chiche, B. N., Woiselle, A., Frontera-Pons, J., & Starck, J. L. (2022). Stable long-term recurrent video super-resolution. In Proceedings of the IEEE/CVF conference on computer vision and pattern recognition (pp. 837-846).

[5] Chen, P., Yang, W., Wang, M., Sun, L., Hu, K., & Wang, S. (2021). Compressed domain deep video super-resolution. *IEEE Transactions on Image Processing*, *30*, 7156-7169.

[6] Shi, S., Gu, J., Xie, L., Wang, X., Yang, Y., & Dong, C. (2022). Rethinking alignment in video super-resolution transformers. Advances in Neural Information Processing Systems, 35, 36081-36093.

[7] Isobe, T., Jia, X., Tao, X., Li, C., Li, R., Shi, Y., ... & Tai, Y. W. (2022). Look back and forth: Video super-resolution with explicit temporal difference modeling. In Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition (pp. 17411-17420).

[8] Qiu, Z., Yang, H., Fu, J., & Fu, D. (2022, October). Learning spatiotemporal frequency-transformer for compressed video super-resolution. In European Conference on Computer Vision (pp. 257-273). Cham: Springer Nature Switzerland.

[9] Xiao, Z., Fu, X., Huang, J., Cheng, Z., & Xiong, Z. (2021). Space-time distillation for video super-resolution. In Proceedings of the IEEE/CVF conference on computer vision and pattern recognition (pp. 2113-2122).

[10] Zhang, A., Li, Q., Chen, Y., Ma, X., Zou, L., Jiang, Y., ... & Muntean, G. M. (2021). Video super-resolution and caching—An edge-assisted adaptive video streaming solution. IEEE Transactions on Broadcasting, 67(4), 799-812.

[11] Yu, J., Liu, J., Bo, L., & Mei, T. (2022). Memory-augmented non-local attention for video super-resolution. In *Proceedings of the IEEE/CVF conference on computer vision and pattern recognition* (pp. 17834-17843).
[12] Zhu, X., Li, Z., Lou, J., & Shen, Q. (2021). Video super-resolution based on a spatio-temporal matching network. Pattern Recognition, 110, 107619.
[13] Hu, M., Jiang, K., Wang, Z., Bai, X., & Hu, R. (2023). Cycmunet+: Cycle-projected mutual learning for spatial-temporal video super-resolution. *IEEE Transactions on Pattern Analysis and Machine Intelligence*.
[14] Wang, L., Hajiesmaili, M., & Sitaraman, R. K. (2021, October). Focas: Practical video super-resolution using foveated rendering. In Proceedings of the 29th ACM International Conference on Multimedia (pp. 5454-5462).
[15] Luo, L., Yi, B., Wang, Z., Yi, P., & He, Z. (2024). Efficient lightweight network for video super-resolution. Neural Computing and Applications, 36(2), 883-896.
[16] Lin, J., Huang, Y., & Wang, L. (2021). FDAN: Flow-guided deformable alignment network for video super-resolution. *arXiv preprint arXiv: 2105.05640*.
[17] Luo, J., Huang, S., & Yuan, Y. (2020, October). Video super-resolution using multi-scale pyramid 3d convolutional networks. In Proceedings of the 28th ACM International Conference on Multimedia (pp. 1882-1890).
[18] Armin Kappeler, Seunghwan Yoo, Qiqin Dai, and Aggelos K. Katsaggelos." Video Super-Resolution with Convolutional Neural Networks". IEEE Transactions on Computational Imaging 2016.
[19] Li, Tianyi, et al. "A Deep Learning Approach for Multi-Frame In-Loop Filter of HEVC." IEEE Transactions on Image Processing 28.11 (2019): 5663-5678.
[20] Liu, Dong, et al. "Deep Learning-Based Technology in Responses to the Joint Call for Proposals on Video Compression with Capability beyond HEVC." IEEE Transactions on Circuits and Systems for Video Technology (2019).
[21] Lin, Hongwei, et al. "Improved Low-Bitrate HEVC Video Coding using Deep Learning based Super-Resolution and Adaptive Block Patching." IEEE Transactions on Multimedia(2019).
[22] Wang, Y., Guo, J., Gao, H., & Yue, H. (2021). UIEC^ 2-Net: CNN-based underwater image enhancement using two color spaces. Signal Processing: Image Communication, 96, 116250.
[23] Magnusson, M., Sigurdsson, J., Armansson, S. E., Ulfarsson, M. O., Deborah, H., & Sveinsson, J. R. (2020, September). Creating RGB images from hyperspectral images using a color matching function. In IGARSS 2020-2020 IEEE International Geoscience and Remote Sensing Symposium (pp. 2045-2048). IEEE.
[24] Alwan, Z. A., Farhan, H. M., & Mahdi, S. Q. (2020). Color image steganography in YCbCr space. International Journal of Electrical and Computer Engineering, 10(1), 202.
[25] Ansari, M., & Singh, D. K. (2022). Significance of color spaces and their selection for image processing: a survey. Recent Advances in Computer Science and Communications (Formerly: Recent Patents on Computer Science), 15(7), 946-956.
[26] Saleem, E., & El Abbadi, N. K. (2020). Auto colorization of grayscale image using YCbCr color space. Iraqi Journal of Science, 3379-3386.
[27] Sahu, M., & Dash, R. (2021). A survey on deep learning: convolution neural network (CNN). In Intelligent and Cloud Computing: Proceedings of ICICC 2019, Volume 2 (pp. 317-325). Springer Singapore.
[28] Kumar, V., Choudhury, T., Satapathy, S. C., Tomar, R., & Aggarwal, A. (2020). Video super resolution using convolutional neural network and image fusion techniques. International Journal of Knowledge-based and Intelligent Engineering Systems, 24(4), 279-287.
[29] Daithankar, M. V., & Ruikar, S. D. (2020). Video super resolution: a review. In ICDSMLA 2019: Proceedings of the 1st International Conference on Data Science, Machine Learning and Applications (pp. 488-495). Springer Singapore.
[30] Khan, A., Sohail, A., Zahoora, U., & Qureshi, A. S. (2020). A survey of the recent architectures of deep convolutional neural networks. Artificial intelligence review, 53, 5455-5516.
[31] Sekehravani, E. A., Babulak, E., & Masoodi, M. (2020). Implementing canny edge detection algorithm for noisy image. Bulletin of Electrical Engineering and Informatics, 9(4), 1404-1410.
[32] Cao, Y., Wu, D., & Duan, Y. (2020). A new image edge detection algorithm based on improved Canny. Journal of Computational Methods in Sciences and Engineering, 20(2), 629-642.
[33] Sidén, P., & Lindsten, F. (2020, November). Deep Gaussian Markov random fields. In International conference on machine learning (pp. 8916-8926). PMLR.
[34] Blake, A., Kohli, P., & Rother, C. (Eds.). (2011). Markov random fields for vision and image processing. MIT press.
[35] Geman, S., & Graffigne, C. (1986, August). Markov random field image models and their applications to computer vision. In Proceedings of the international congress of mathematicians (Vol. 1, p. 2).
[36] Kim, J., J. K. Lee, and K. M. Lee. "Accurate Image Super-Resolution Using Very Deep Convolutional Networks." Proceedings of the IEEE® Conference on Computer Vision and Pattern Recognition. 2016, pp. 1646-1654.
[37] Xue, T., Chen, B., Wu, J., Wei, D., & Freeman, W. T. (2017). *Video Enhancement with Task-Oriented Flow*. arXiv.
[38] Zamzam, P., Rezaei, P., Khatami, S. A., & Appasani, B. (2025). Super perfect polarization-insensitive graphene disk terahertz absorber for breast cancer detection using deep learning. Optics & Laser Technology, 183, 112246.
[39] Cao, Y., Wang, C., Song, C., Tang, Y., & Li, H. (2021, July). Real-time super-resolution system of 4k-video based on deep learning. In 2021 IEEE 32nd International Conference on Application-specific Systems, Architectures and Processors (ASAP) (pp. 69-76). IEEE.
[40] Pan, J., Bai, H., Dong, J., Zhang, J., & Tang, J. (2021). Deep blind video super-resolution. In Proceedings of the IEEE/CVF international conference on computer vision (pp. 4811-4820).
[41] Li, F., Bai, H., & Zhao, Y. (2020). Learning a deep dual attention network for video super-resolution. IEEE transactions on image processing, 29, 4474-4488.
[42] Isobe, T., Li, S., Jia, X., Yuan, S., Slabaugh, G., Xu, C., ... & Tian, Q. (2020). Video super-resolution with temporal group attention. In Proceedings of the IEEE/CVF conference on computer vision and pattern recognition (pp. 8008-8017).
[43] Bai, H., & Pan, J. (2024). Self-supervised deep blind video super-resolution. IEEE Transactions on Pattern Analysis and Machine Intelligence.
[44] Feng, Z., Zhang, W., Liang, S., & Yu, Q. (2023). Deep video super-resolution using a hybrid imaging system. IEEE Transactions on Circuits and Systems for Video Technology, 33(9), 4855-4867.
[45] Wang, W., Liu, Z., Lu, H., Lan, R., & Zhang, Z. (2023). Real-Time Video Super-Resolution with Spatio-Temporal Modeling and Redundancy-Aware Inference. Sensors, 23(18), 7880.