Mesopotamian journal of Cybersecurity
Vol.4,No.3, **pp**. 318–334
DOI: https://doi.org/10.58496/MJCS/2024/029; ISSN: 2958-6542
https://mesopotamian.press/journals/index.php/cybersecurity

Research Article

From Monochrome to Color: Efficient Techniques for Realistic Video Colorization

Zahoor M. Aydam [1,2*], Nidhal K. El Abbadi [3]

[1]*Computer Science Department, Faculty of Computer Science and Mathematics, University of Kufa, Najaf, Iraq*

[2]*Computer Sciences and Mathematics College, University of Thi_Qar, Iraq*

[3] *Al-Mustaqbal Center for AI Applications, Al-Mustaqbal University, Babylon, Iraq*

ARTICLEINFO

Article history

Received 07 Oct 2024
Accepted 08 Dec 2024
Published 29 Dec 2024

Keywords

Grey video
ELU
reference image
Trilinear up sampling
encoder and decoder
BN
Lab
RGB
PSNR
SSIM

ABSTRACT

Colorizing grayscale videos is a challenging task that involves adding colors to monochromatic videos to make them appear as natural and realistic as possible, despite the absence of information about the original color distribution. This task is crucial for applications such as restoring historical videos and creating media content requiring realistic and accurate colorization. However, existing methods often face issues such as poor temporal stability, color distortions, and the need for extensive post processing under certain conditions. To address these challenges, this study proposes a novel approach comprising a pre-processing network and a source-reference network trained in an end-to-end manner. The pre-processing network employs an encoder-decoder architecture enhanced with temporal convolutions and skip connections, enabling it to improve video quality, adapt to resolution changes, and leverage batch normalization (BN) and exponential linear unit (ELU). The source-reference network incorporates an encoder for reference image processing and a fusion module with residual blocks to combine feature maps through a source-reference attention mechanism. The final output is generated in the Lab color space and converted to the RGB format, ensuring high-quality video colorization with enhanced temporal stability. Experimental evaluations demonstrate that the proposed model achieves significant improvements in color realism and temporal stability, with a PSNR of 37.89 and an SSIM of 0.999982—which surpasses those of most state-of-the-art methods. These results confirm the effectiveness and applicability of the proposed method for video colorization tasks, making it a robust solution in the domain.

1. INTRODUCTION

Automated colorization refers to the process by which black-and-white images or videos are digitally transformed into color via artificial intelligence. This technique integrates advanced color vision algorithms with high-level image processing to achieve realistic and visually appealing results.

Its importance is particularly evident in modern photography and filmmaking, where contemporary aesthetics demand diverse visual styles and interpretive techniques. Historically, the concept of colorization dates back to early black-and-white movies and television programs [1]. This historical context highlights the ongoing evolution of colorization techniques to address new technological and creative needs.

Despite significant advancements, video colorization still presents difficulties. Key issues include ensuring the temporal stability of color changes across frames and managing the computational complexity of video data, which is far greater than that of still images. A persistent challenge is achieving coherent colors across frames while accounting for object movements and dynamic scene changes [2]. Existing methods that optimize still images often fail to address temporal coherence comprehensively in videos [3]. This gap emphasizes the necessity for innovative approaches that ensure accurate color deposition and stable transitions in motion sequences.

This research proposes a novel framework that addresses these challenges by introducing mechanisms for precise colorization of successive frames, ensuring smooth transitions between scenes, and managing diverse motion patterns. The proposed method is fully automated and minimizes human intervention, enhancing efficiency and usability [4].

*Corresponding author. Email: zahoorm.alfraih@student.uokufa.edu.iq

The primary objective of this research is to develop a robust framework for video colorization that achieves temporal stability, reduces color distortions, and handles video data efficiently. These improvements aim to overcome the limitations of traditional methods and provide an effective solution for large-scale applications.
The outcomes of this study have broad implications for enhancing the realism and quality of visual media. By addressing fundamental challenges in video colorization and offering a comprehensive solution, this research contributes to historical video restoration, media production, and artistic endeavors [5][6]. This unified approach not only improves current practices but also sets a foundation for future advancements in automated colorization techniques.
The remainder of the paper is organized as follows: Section Two reviews relevant previous studies. The theoretical background of the algorithms used in this work is presented in Section Three, followed by a description of the quality metrics employed in this study in Section Four. Section Five provides a detailed explanation of the proposed method. The results and discussion are presented in Section Six, and Section Seven concludes the paper.

2. RELATED WORK

K., Tejashwini, et al. A fully automated version of the colorization method was introduced; the task was formulated as a classification problem to increase the range of colors used. To address the nature of the task as not fully deterministic at the time when the training data were generated, class-rebalancing measures were used. In the testing phase, the system uses a CNN in a feed-forward structure after it has been trained on a database containing more than a million color images. To make the evaluation, the method accomplished the so-called “colorization Turing test”, and the result was that it cheated people in 32% of the cases, which was better than previous technologies did. Additionally, the approach proves to be useful as a pertaining task of self-supervised learning, which demonstrates competitive results in feature learning evaluation metrics. However, the method is best suited for single-image colorization, while temporal consistency within video sequences is lacking. This is because the approach fails to consider the order of consecutive frames, which is crucial for preserving color consistency across frames where these videos contain movements. Further development of these strategies for processing temporal dependencies would help address this flaw in methods that make it possible to colorize video sequences, although quite well, but not very smoothly and coherently [7].

Guangzi, Zhang et al. developed a variational autoencoder framework for the specific task of colorizing videos via spatiotemporal information to improve the colorization of black and white frames within the video sequence. To enhance temporal coherence, the method synchronizes semantic similarity with color spreading, directing colorization across the frames of a given timeline. The framework can convert a video frame to a desired different-dimensional representation space through an encoder network, whereas a decoder network translates the resulting latent features into RGB color channel images. In contrast to prior approaches, this strategy enables adequate treatment of temporal dependencies between frames and constant and accurate colouration of the video frames. Moreover, a specific loss function is used to enhance conductivity in terms of spatial and temporal qualities; there are significant improvements in video colorization. However, the method is still not completely adequate for handling more complex and rapidly changing scenes [8].

Yizhang Yang et al. proposed a video colorization method using a reference-based model to solve these issues. Unlike exemplar-based approaches that completely depend on color details in the reference frame only, this model adds an AB chrominance point to compensate for missing color data, hence increasing the color accuracy. Moreover, to utilize information generated from warped images, a bidirectional optical flow propagation network is integrated, further improving temporal smoothing across frames. The experimental outcomes show that the proposed method is superior to the most closely related benchmarks in terms of colorization quality and consistency across datasets. While this model improves temporal consistency and color accuracy to a great degree, it remains frame-dependent—the quality of the reference frame. When the reference frame is unhelpful in dynamic scenes, the model may fail to sustain high accuracy in color [9].

Siqi, Chen, et al. proposed a video colorization approach that incorporates references and long-term spatiotemporal contextual information. The method introduces a CNN-transformer block to enhance spatial dependencies by integrating local texture and structural features, along with a double-head nonlocal operation to refine feature performance. For temporal consistency, a linkage subnet is used to transfer motion details between consecutive frames, improving frame-to-frame transitions. Experiments show that the approach achieves vibrant colorization and reduced motion blur, outperforming existing methods. However, challenges remain in handling rapid motion and frequent scene changes, where the linkage subnet struggles to maintain continuity. Expanding the temporal context or incorporating advanced motion analysis could address these limitations, enabling more realistic and stable colorization for highly dynamic video sequences. The method struggles with fast-moving and complex scenes, limiting its ability to deliver consistent colorization under such conditions. Enhancements in motion analysis and temporal context are needed to bridge this gap [10].

3. THEORETICAL BACKGROUND

The theoretical background encompasses a variety of algorithms that together contribute to the achievement of the desired objective. These algorithms include the following:

3.1 Residual blocks

A fundamental residual block combines a simple structure, which ensures high speed, with a deep neural network, which is desirable for the block. The proposed method incorporates this architecture because it has the potential to manage the vanishing gradient problem, thus allowing gradients to propagate throughout the network. However, residual blocks help networks retain identity mapping as well as other corrections that help in generalization in addition to other corrections that enable the model to learn other patterns. Fig. 1 illustrates the block structure, which comprises the following steps:

- A series of layers consists of convolutional layers that are used in the task of feature extraction from the input. Typically, in a residual block, two layers of convolution with a kernel size of 3 × 3 and a padding of 1 are ned. The number of layers, the dimensions of the kernel, and other parameters can be set according to the application or the problem [11][13].
- Batch normalization: Batch normalization, which is used after each convolutional layer, is also a critical feature, as it helps improve and speed up the training process through normalization of the input data for the next layer. Its main goal is to solve a problem called internal covariate shift, which results from fluctuations in the input distribution to a layer during training[14].
- Nonlinear activation functions, especially the rectified linear unit (ReLU), are widely used in residual blocks. Like batch normalization, ReLU is applied after the first and positioned after the block after the skip connection. The introduction of nonlinearity is another primary benefit when ReLU is used since it helps in obtaining complex and nonlinear data patterns in a given set[15].
- Skip (shortcut) connections: The basis of the design of residual blocks is the use of connections between two blocks called shortcut connections. These connections are used to add the input of the block to the output of the last batch normalization layer. In this way, gradients can pass more effectively through a network and therefore solve the vanishing gradient problem and increase or extend the depth of the neural network.

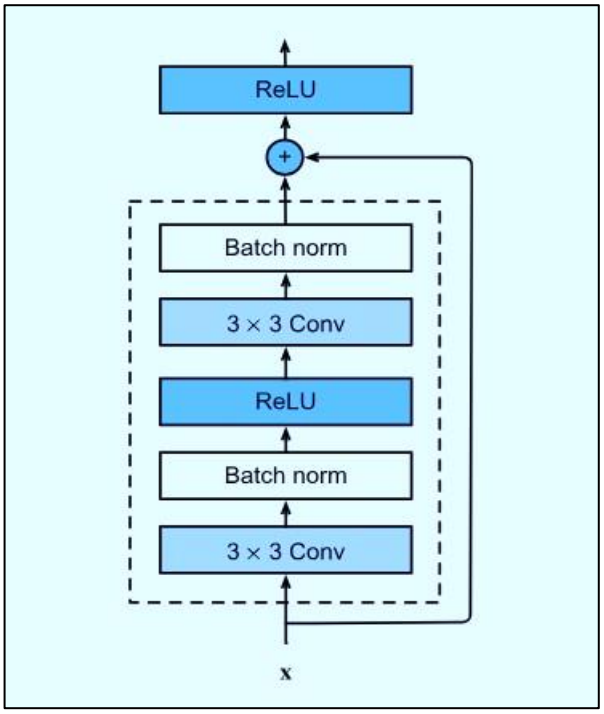

Fig. 1. Residual block

In the residual block, before applying the activation function to the output of this layer, the input given to the block is concatenated with the output generated by these layers. This ensures that the neural network can memorize not only the identity function but also any corrections that need to be made to the entire loss, which helps the network to generalize and understand other higher-order patterns [12].

3.2 Encoder-Decoder Network

An encoder-decoder is a type of neural network model that is particularly applicable to sequence-to-sequence learning models, such as machine translation and image captioning. It comprises two main components: the encoder and the decoder, as illustrated in Fig. 2. The encoder transforms the input sequences to generate the context vectors, and the decoder makes use of the context vector to create output sequences. The encoding process encompasses the process of transforming input data into a quantitative form while preserving the interrelationships of the data structure. The decoder, on the other hand, reconstructs the encoded representation back into its original form or a similar output. To achieve this, attention mechanisms are often employed to establish relationships between the encoded data and the reconstructed output, enhancing the accuracy of the generated sequences [15].

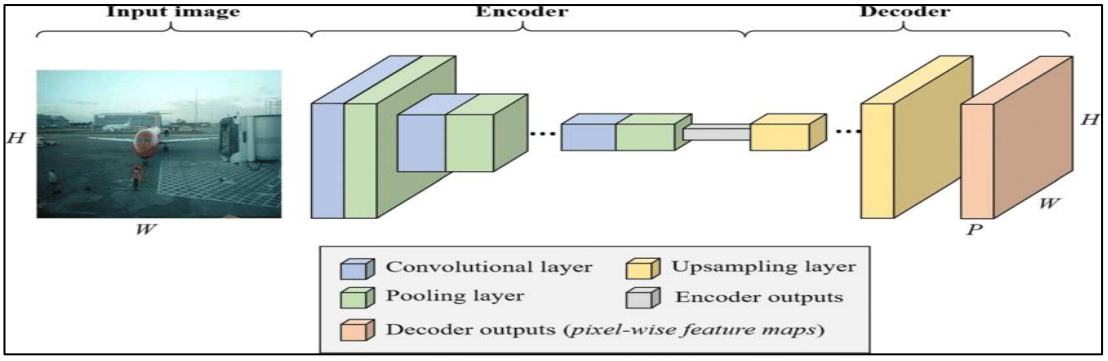

Fig. 2. Encoder-decoder architecture

3.3 Multi-Layer Feature Fusion for Local and Global Features (MFLL)

In the development of this proposal, we have outlined two approaches to use MFLL on the features of CNNs of various levels. The first is known as intralayer fusion, which attempts to increase the number of local components of the CNN. In the next layer of the architecture, the strengthened features are delivered. The improvement of features' partial information enhances the ability of CNNs to extract feature details. As shown in Fig. 3, the feature maps of Layer1 and Layer2 include several convolution sublayers. The module subsequently transmits the result to Layer 3 and therefore improves the local feature information of Layers 1 and 2. In general, it is possible to utilize several types of modules to enhance these or other local characteristics.

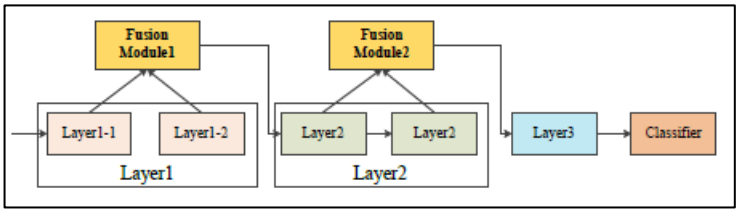

Fig. 3. Two intralayer fusion modules work to enhance different parts of the local parts.

The second approach is to improve the features of the global information, known as interlayer fusion. While forward propagating the CNN, some feature information is pooled, and multilayer feature fusion combines the feature information of the previous layers into the fused feature map through interlayer fusion and fuses the global features, enhancing the diversity of the final feature maps. Hence, it can reclaim and/or offset the loss of the feature. Fig. 4 shows that the module takes feature maps from three different groups of convolution layers. The module then compiles them and sends them to a classifier to obtain a classification result. Indeed, it is often a single fusion module that is applied in the CNN[16].

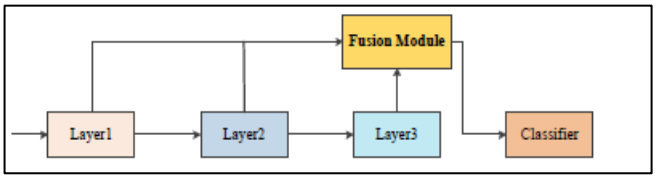

Fig. 4. Interlayer fusion to enhance the global feature

3.4 Exponential linear unit

The exponential linear unit (ELU) is a type of activation function designed to increase the training efficiency of deep neural networks. ELUs mitigate the vanishing gradient problem by acting linearly for positive inputs and allowing negative outputs, which brings the mean activation closer to zero. This reduction in computational complexity aids in training the model more quickly. This is because ELUs have properties such as centring the activations around zero and include a saturation plane for negatives as opposed to ReLU and its variants such as leaky ReLU. These characteristics allow ELUs to train models with better representations and improve the learning rate and generalizability, especially in neural networks that are deeper than five layers, making them offer state-of-the-art performance in different applications [17].

The ELU is mathematically expressed by the following equation: the exponential linear unit (ELU) is defined as

$$f(q) = \begin{cases} q, & q \geq 0 \\ \alpha * \exp(q) - 1, & q < 0 \end{cases} \quad (1)$$

3.5 Trilinear up sampling

In CNNs, feature map resolution can be increased through trilinear up sampling of feature maps. This is especially helpful in operations requiring 3D data as input, including medical volumetric imaging or video frame processing. The up sampling process consists of the proper interpolation of the feature map values to obtain a new feature map with a higher resolution[18].

Concerning CNNs, trilinear up sampling is a certain form of trilinear interpolation usually employed in operations that gradually introduce spatial dimensions through several steps or layers. The trilinear up sampling in the CNN involves the following:

- **Feature Map Preparation:** The input feature map is described in terms of height, width, and channel, whereas the depth dimension can be considered another spatial dimension in trilinear interpolation.
- **Grid creation:** In the up sampled feature map, new pixel values need to be predicted, and a grid of points is formed for this purpose.
- **Interpolation:** Trilinear interpolation is performed at every point in the up sampled grid (calculating pixel values from nearby known pixels of the original feature map). The interpolation determines the value of each point on the basis of how far it is from nearby points in all three dimensions.
- **Weighting and Summing:** The weights are determined by the distance of a new point to known points in that feature map. A weighted sum of the points with some weight on the basis of the distance from the Tuple to predict its value

3.6 Source-reference attention layer

The source-reference attention layer, also known as the cross-modal attention layer or cross-modal interaction layer, is an important component in neural networks that enables the incorporation of information from reference feature maps into source feature maps. It is especially effective in those cases when data from another but connected input (the reference) can be beneficial in representing the primary input (the source) [19].

Steps detail how the source–reference attention layer functions[20]:

- **Projection into Common Feature Space:**
  - The source and reference feature maps are linearly transformed to create three different projections: Queries (Q), Keys (K), and Values (V).
  - Queries (Q): Projected from source feature maps via a function fq, typically a convolution or linear layer.

$$Q = fq(hs) \quad (2)$$

  - Keys (K) and Values (V): Projected from reference feature maps via functions f_k and fv.

$$K = fk(hr) \quad (3)$$

$$V = fv(hr) \quad (4)$$

  - All projections (Q, K, V) typically share a common dimension d.
- **Attention Score Computation:**

- The similarity scores (attention scores) between each pair of projected source and reference features are computed. This can be done via a dot product:

$$A_{ij} = Q_i . K_j^T \quad (5)$$

- These scores are normalized to obtain weights that sum to 1 across the reference dimension. This is usually achieved via the Softmax function:

$$\alpha_{ij} = \frac{\exp(A_{ij})}{\sum_j \exp(A_{ij})} \quad (6)$$

- **Weighted aggregation:**
  - The attention scores are used to compute a weighted sum of the reference features (V) for each source feature location:

$$h'_{si} = \alpha_{ij} V_j \quad (7)$$

  - This step integrates the relevant information from the reference feature maps into the source feature maps.
- **Combining with Source Features:**
  - The transformed source features (hs') are combined with the source features (hs). This combination is often performed via elementwise addition:

$$h''_s = h'_s \oplus h_s \quad (8)$$

  - This operation preserves the local information from the source features while enhancing it with the contextual information from the reference features.

4. QUALITY METRICS

In the realm of image processing algorithm development, video quality measurement (VQM) plays a significant role. The use of VQM is crucial in assessing the efficacy of processed videos. Video quality denotes a property of a video that gauges the degradation of the processed video to an ideal video [21][22]. The tools used in this proposal to measure model performance are as follows:

4.1 Peak signal-to-noise ratio (PSNR)

This metric has been extensively utilized in numerous image and video processing applications for the quantitative assessment of the similarity or dissimilarity between two images or videos (modified and original) of identical structure. This metric is specified through Eq. (9).

$$PSNR = 10 \log_{10} \left(\frac{M^2}{MSE} \right) \quad (9)$$

The determination of the value of M is contingent upon the structure of the image or video being utilized. In the case of an 8-bit image, the value of M is set at 255. An uphigh value of the peak signal-to-noise ratio (PSNR) indicates that a reconstruction represents higher quality.

4.2 Average Difference (AD)

It is used to calculate the average change between the resulting colored image and the original image. AD can be mathematically expressed by Eq. (10).

$$AD = \frac{1}{XY} \sum_{p=1}^{X} \sum_{q=1}^{Y} [K(p,q) - Q(p,q)] \quad (10)$$

Zero is the optimal numerical value for AD.

4.3 Root mean square error (RMSE)

Similar to the mean squared error, but with the additional step of taking the square root of the result.

$$RMSE = \sqrt{\frac{1}{XY} \sum_{p=1}^{X} \sum_{q=1}^{Y} (K(p,q) - Q(p,q))^2} \quad (11)$$

A smaller numerical value of the root mean square error (RMSE) indicates a better outcome.

4.4 Maximum difference (MD)

The maximum error signifies the difference between the original image and the image after coloring.

$$\mathrm{MD} = \max|\mathrm{K(p,q)} - \mathrm{Q(p,q)}| \tag{12}$$

The image quality decreases as the MD value increases.

4.5 Structural Content (SC)

It computes how similar two images are

$$\mathrm{SC} = \frac{\sum_{p=1}^{X}\sum_{q=1}^{Y}(Q(p,q))^2}{\sum_{p=1}^{X}\sum_{q=1}^{Y}(K(p,q))^2} \tag{13}$$

A higher value of SC indicates that the image is of poor quality.

4.6 Normalized absolute error (NAE)

It also computes how similar two images are

$$\mathrm{NAE} = \frac{\sum_{\mathrm{p=1}}^{\mathrm{X}}\sum_{\mathrm{q=1}}^{\mathrm{Y}}|\mathrm{K(p,q)-Q(p,q)}|}{\sum_{\mathrm{p=1}}^{\mathrm{X}}\sum_{\mathrm{q=1}}^{\mathrm{Y}}\mathrm{K(p,q)}} \tag{14}$$

The highest-quality image is achieved when the NAE approaches zero.

4.7 Normalized Cross-Correlation (NCC)

The measure indicates the contrast between the modified image and the original image. The model is presented as follows:

$$\mathrm{NCC} = \sum_{p=1}^{X}\sum_{q=1}^{Y}\frac{K(p,q)*Q(p,q)}{(K(p,q))^2} \tag{15}$$

4.8 Structure Similarity Index (SSIM)

The SSIM index assesses how similar a tested image X is to the original image Y visually via a formula.

$$\mathrm{SSIM(p,q)} = [K(p,q)]^{\alpha}.[Q(p,q)]^{\beta}.[W(p,q)]^{\gamma} \tag{16}$$

where α, β, and γ are the parameters that determine the significance of each component.

$$\mathrm{K(p,q)} = (2*\mathrm{M_p}*\mathrm{M_q}+\mathrm{S_1})/(\mathrm{M_p^2}+\mathrm{M_q^2}+\mathrm{S_1}) \tag{17}$$

$$\mathrm{Q(p,q)} = (2*\sigma_{\mathrm{p}}*\sigma_{\mathrm{q}}+\mathrm{S_2})/(\sigma_{\mathrm{p}}^2+\sigma_{\mathrm{q}}^2+\mathrm{S_2}) \tag{18}$$

$$\mathrm{W(p,q)} = (\sigma_{\mathrm{pq}}+\mathrm{S_3})/(\sigma_{\mathrm{p}}*\sigma_{\mathrm{q}}+\mathrm{S_3}) \tag{19}$$

where the constants S_1, S_2 and S_3 are utilized to prevent any instabilities that may average from the pixel value ($M_p^2 + M_q^2$), standard deviation ($\sigma_p^2 + \sigma_q^2$) or ($\sigma_p * \sigma_q$) approaches zero. The SSIM (p, q) varies between 0 (indicating dissimilarity). 1 (representing identical patches).

4.9 Pearson correlation coefficient (PCC)

The utilization of a statistical formula known as the Pearson correlation coefficient (PCC) is instrumental in assessing the strength of the relationship between two variables. This formula yields a value that falls within the range of [-1, 1].

$$\mathrm{PCC} = \frac{\mathrm{p}(\sum \mathrm{KQ}) - \sum \mathrm{K}*\sum \mathrm{Q}}{\sqrt{[\mathrm{p}\sum \mathrm{K^2} - (\sum \mathrm{K})^2]*[\mathrm{p}\sum \mathrm{Q^2} - (\sum \mathrm{Q})^2]}} \tag{20}$$

A positive value of the correlation coefficient signifies a positive correlation between the variables, whereas a negative value implies a negative correlation between the variables.

5. PROPOSED METHOD

The proposed approach for video colorization is structured into three main phases: preprocessing, reference image processing, and source-reference network application. These phases march together to handle images and videos within the shortest time possible without considering the size of the image or the video. Below is a more detailed and clarified breakdown of each phase.

5.1 First phase: Preprocessing

The main objective of the preprocessing step is the process of eliminating the input monochrome video noise in addition to making adjustments to the video before the colorization step begins. This phase guarantees that the frames used in the production of the actual video have been designed to provide accurate colorization. The process is implemented in an encoder–decoder framework with eight convolutional layers, four in the encoder section and four in the decoder section. This is followed by batch normalization and an exponential linear unit activation function for each convolutional layer. The different phases of the many stages involved in preprocessing are shown in Fig. 5.

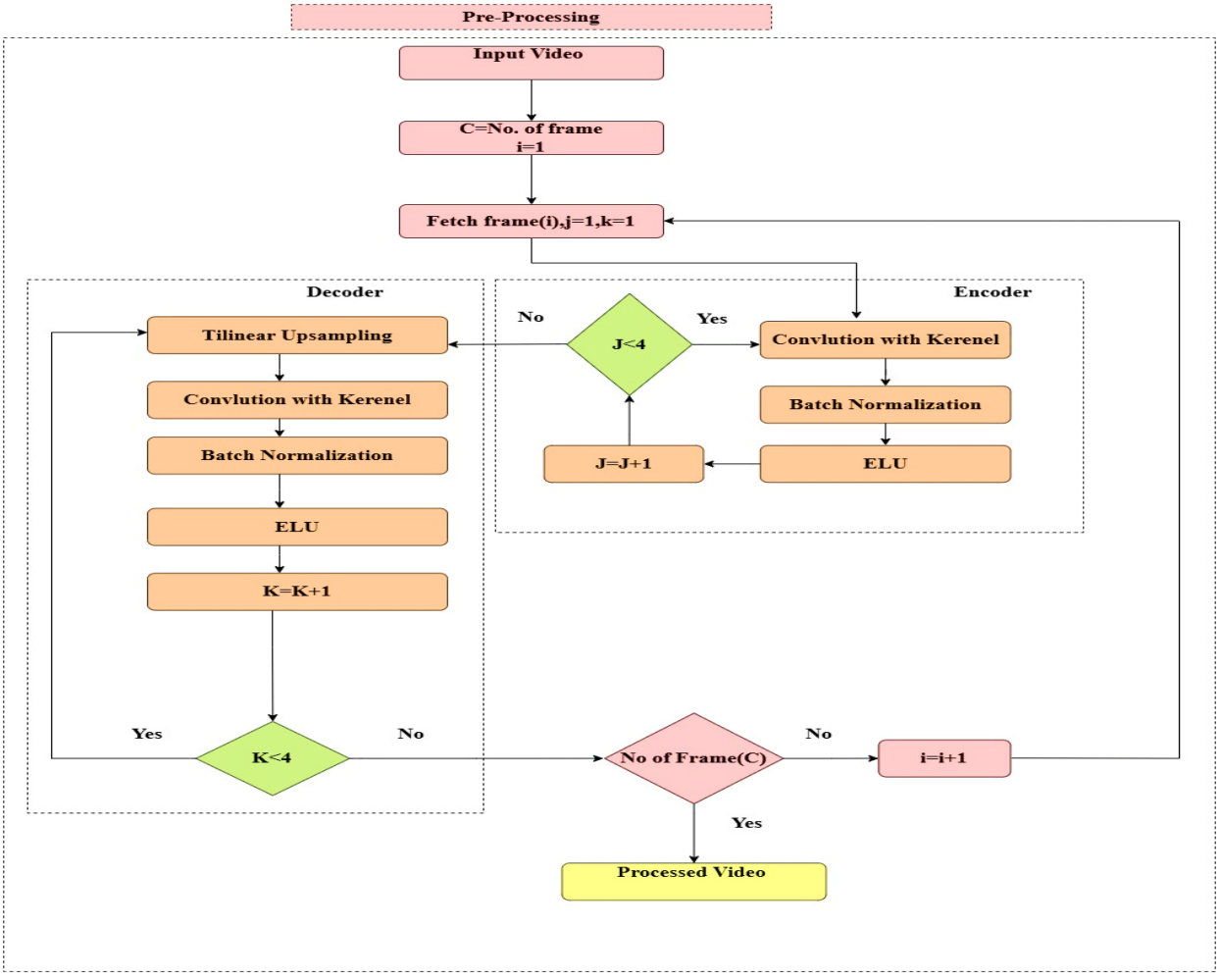

Fig. 5. Flowchart of the proposed method at the preprocessing stage.

Preprocessing consists of three main stages:

- **First stage: Encoding**

While encoding, every frame in a video is subjected to feature extraction via 3 by 3 convolutional layers. To maintain equal dimensions of the input and output, padding is reproduced, and a step of 2 pixels is used to reduce the frames' resolution without losing their important characteristics.

- **Second Stage: Decoding**

During decoding, the features from the encoding stage are again passed through some transformations to reconstruct the frames of the video. The strategy implemented here is trilinear up sampling and 3×3 convolution but replicates padding. There is a 1-pixel stride used to fine-tune the resolution, whereas connections between equal stages of the encoder and decoder networks minimize the noise in the up sampling step and focus on preserving the more relevant importance.

- **Third Stage: Output**

After processing all frames of the video, it outputs a video that can be considered temporally coherent and that is ready for the next colorization step.

5.2 Second phase: Reference image processing

Reference image processing consists of seven stages, starting with the input of reference images and ending with the final processed images. This phase aims to unify and enhance the features of the reference images. The entire process of this phase is illustrated in Fig. 6.

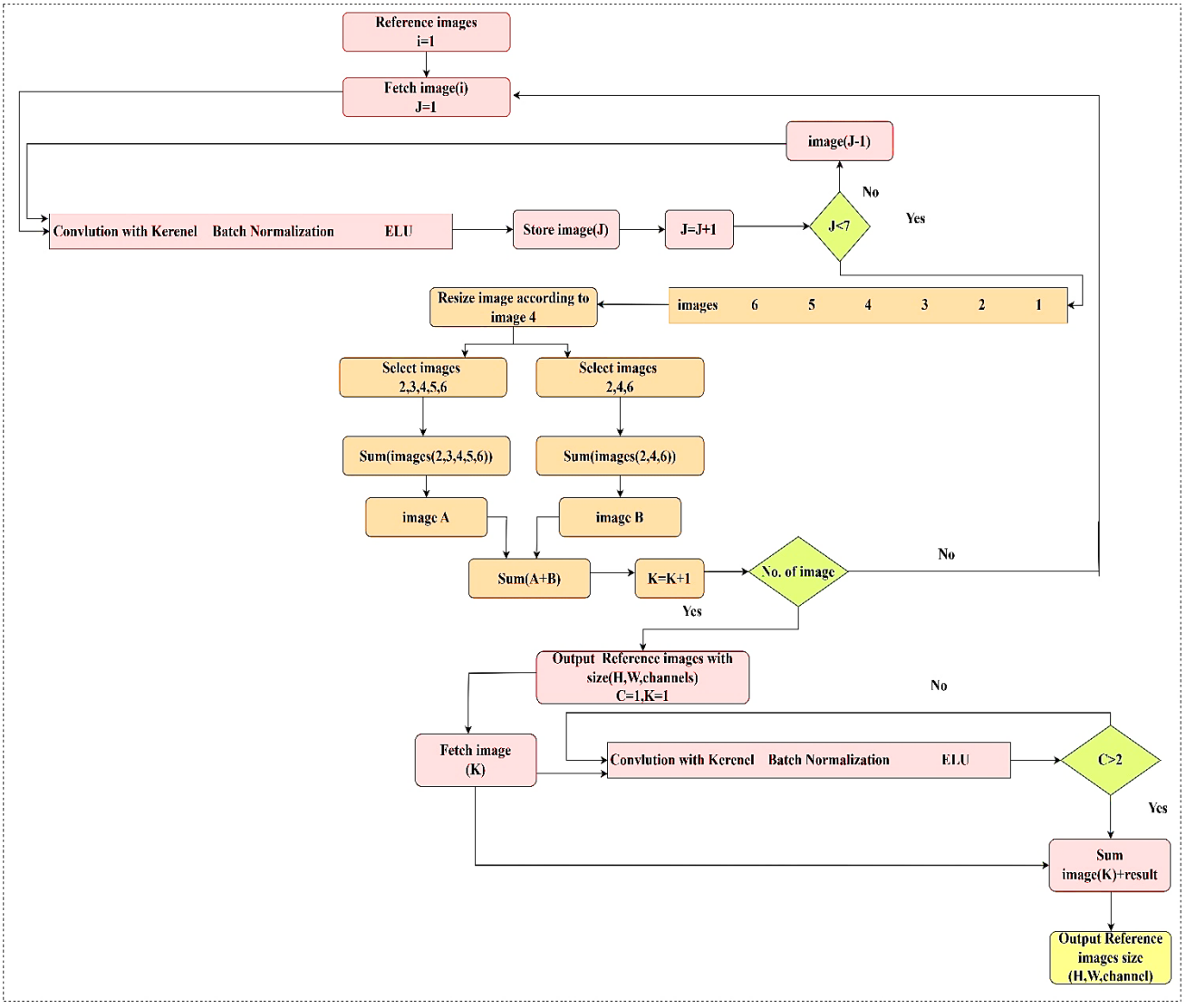

Fig. 6. Process reference images

The stages of reference image processing are as follows:

- **First Stage: Initialization**
  In this stage, the reference image(s) are input into the encoder to start processing.
- **Second Stage: Encoding Processing**
  Each reference image passes through six layers, where each layer includes the following steps:
  - **3x3 Convolution:** Convolution is applied via a 3x3 kernel with a stride of 2 to reduce the dimensions of the image and replicate padding to maintain consistent input and output sizes.

- Batch normalization: **The output from the convolution layer is batch normalized to speed up and stabilize the training process.**
- **ELU Activation**: The outputs are then activated via the ELU function to improve the learning capability and avoid the vanishing gradient problem.

- **Third Stage: Feature Map Unification**
  After extracting the feature maps from each layer, their sizes are unified to match the size of the image produced by the fourth layer. This ensures consistent dimensions across all layers for the reference images.
- **Fourth Stage: Multilayer Feature Fusion**
  Both local and global features are combined through a specialized module in the fusion process, which includes the following steps:
  - **Intralayer Fusion:** Feature maps from successive layers (second, third, fourth, fifth, and sixth) are combined to produce an image (A).
  - **Interlayer Fusion:** Feature maps from non-consecutive layers (second, fourth, and sixth) are combined to produce an image (B).
  - **Combining Images, A and B:** Images (A) and (B) are summed to produce a combined feature map.
- **Fifth Stage: Repeating the Process for All Reference Images**

Each reference image is processed through the aforementioned stages to ensure feature extraction and preparation for the next stage.

- **Sixth Stage: Residual Block Processing**
  After feature fusion, the reference images are fed into the residual blocks in two steps:
  - **Residual Block 1**: Includes convolution, batch normalization, and ELU activation to enhance the features.
    - **Convolution:** Convolutional layers are applied to extract essential features from the reference images.
    - **Batch normalization (BN):** BN normalizes the activations to stabilize and accelerate the training process.
    - **Exponential Linear Unit (ELU) Activation**: Introducing nonlinearity to help the network learn complex patterns.

    This block enhances the initial features extracted from the reference images.
  - **Residual Block 2:** This block repeats the same steps as residual block 1: convolution, batch normalization, and ELU activation, in addition to further refining the features and improving their quality and accuracy. The use of residual blocks with these operations ensures that the features extracted from the reference images are not only enhanced but also further refined, leading to more accurate and robust results in the subsequent processing phases.
- **Seventh Stage: Final output**
  After all the reference images are processed through the residual blocks, the final processed reference images with dimensions (H, W, channels) are obtained.

5.3 Third phase: Source-Reference Network

The source-reference network integrates the color information from the reference images into the processed video frames. This phase is crucial for achieving accurate colorization while maintaining temporal coherence. The entire process is visually represented in Fig. 7, which clarifies the sequential stages involved.

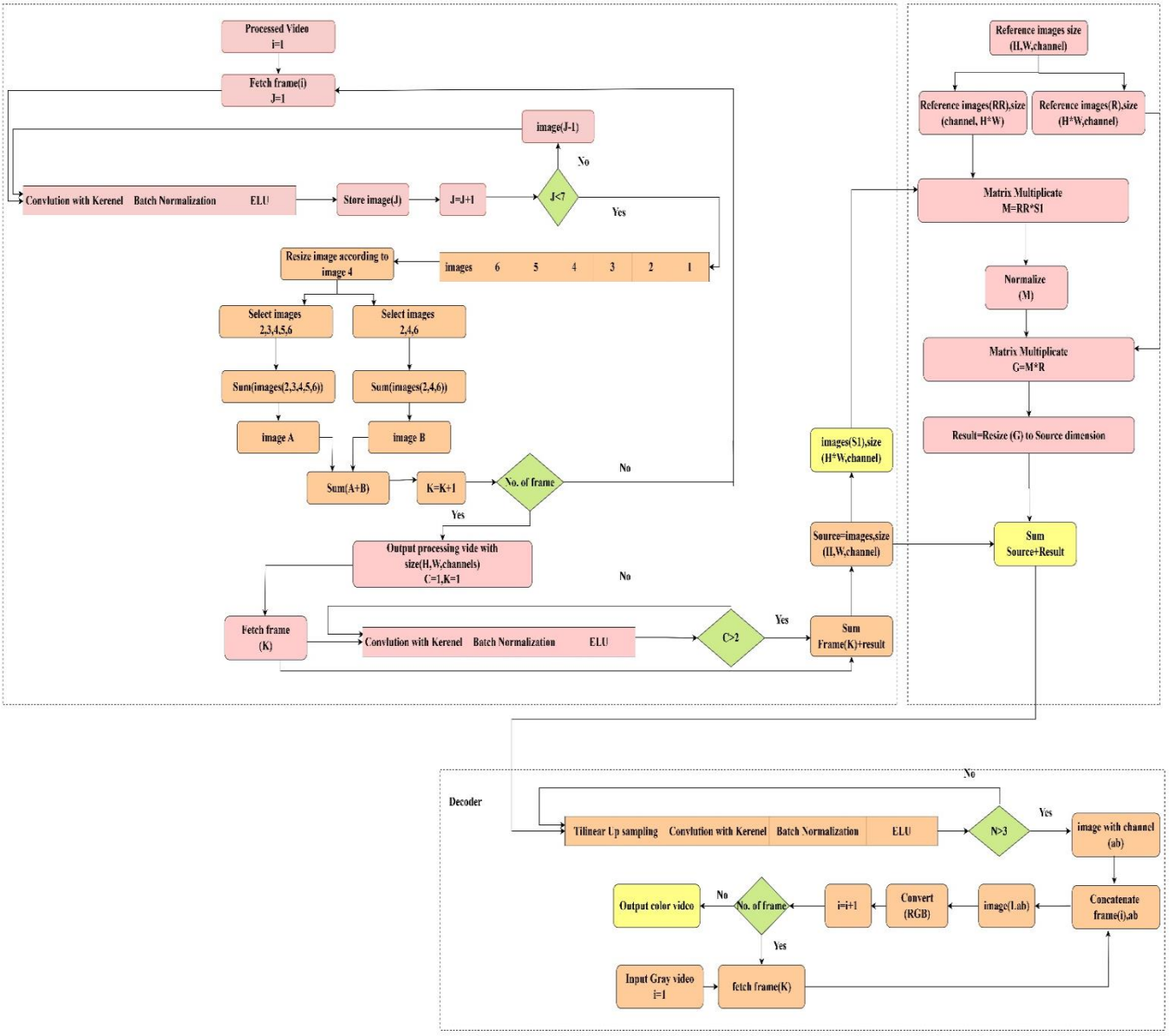

Fig. 7. Process Video and Colorization

The source-reference network consists of three main stages:

- **First Stage: Video Frame Processing**
  After noise removal and enhancement in the preprocessing phase, the video is divided into frames. These frames are processed similarly to the reference images to match their size and features, ensuring compatibility with the colorization stages.
- **Second Stage: Source-Reference Attention**
  The input to this stage consists of both the processed video frames from the first stage and the processed reference images from the second phase. These are fed into the source-reference attention layer. The process begins by converting the reference image(s) into two different sets of dimensions: one with dimensions (H*W, Channel) and the other with dimensions (Channel, H*W). Similarly, the processed video frames are transformed into frames with dimensions (H*W, Channel).
  Next, the reference images with dimensions (channel, H*W) are multiplied by the video frames with dimensions (H*W, channel) (matrix multiplication) to compute attention weights. These attention weights are then normalized via the Softmax function. After normalization, the attention weights are multiplied by the reference images in the dimensions (H*W, Channel) to compute the transformed source feature maps. Finally, the

transformed source features are combined with the frame features of dimensions (H, W, Channel). After processing a frame, the next frame is fed into the source-reference attention layer, and the process is repeated until all frames are processed.

Once all frames have been processed via the first reference image, the next reference image is input into the source-reference attention layer, and all frames are processed again. This cycle continues until all frames have been processed using all reference images.

Following this merging process, the features are subjected to further processing through the source-reference attention layer to ensure temporal consistency before being converted into chrominance channels. Each frame is then fed into the decoder stage, which consists of three layers. Each layer involves the following steps: applying trilinear up sampling, performing convolution with kernel sizes of 3x3, applying replicate padding to ensure consistent input–output sizes with strides of one pixel, and applying batch normalization and the ELU activation function. After processing a frame, the next frame is fed into the decoder, and the process is repeated until all frames are processed.

- **Third Stage: Processing and Output**
  After the source-reference attention stage, the features are further processed to ensure temporal consistency. A video with color channels (A and B) in the Lab color space is produced, which is then combined with the luminance channel (L) from the denoised and quality-enhanced grayscale video in the preprocessing network. The final step involves converting the video from the Lab color space to the RGB format, resulting in a fully colored video that maintains color accuracy and temporal coherence.

6. RESULTS AND DISCUSSION

The model for video colorization in this study is trained and tested on the YouTube-8 M dataset. Various scenes from the dataset were sampled, and in total, 1,569 videos (10,243,010 frames) were carefully annotated to exclude irrelevant content, such as gameplay and monochrome footage. The data were then partitioned to enable structured model development. For training the model, 1,219 videos with 7,993,132 frames were utilized. For the validation process, 50 videos with 321,306 frames were used. For testing, 300 videos with 1,928,572 frames were used. This segmentation provided a good foundation for the training process, parameter adjustment, and overall assessment of model performance concerning generalization. The specifications used to train the models included 100 epochs, a batch size of 16, and a frame size of 224 × 224.

Various tests were conducted to evaluate the efficacy of the proposed algorithm. Fig. 8 illustrates the outcomes of colorizing a set of grayscale frames and compares the peak signal-to-noise ratios (PSNR) of these colorized frames to those of the ground truth images. The PSNR values fall within an acceptable range for the colorization process, indicating the algorithm's effectiveness. Additionally, there is a notable visual resemblance between the ground truth and colorized images. These results are considered very promising for blind colorization, demonstrating the algorithm's ability to produce high-quality colorized frames without prior color information.

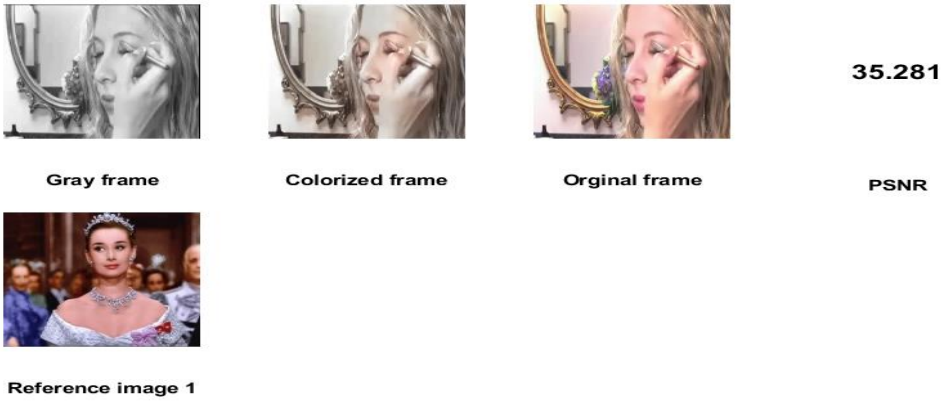

Fig. 8. (a) Grayscale Frame Colorization Using a Reference Image

Fig. 8. (b) Greyscale frame colorization via two reference images

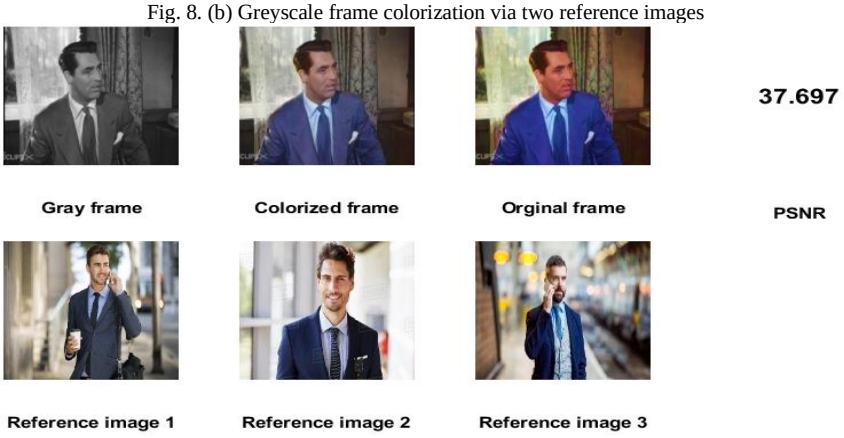

Fig. 8. (c) Grayscale frame colorization using three reference images

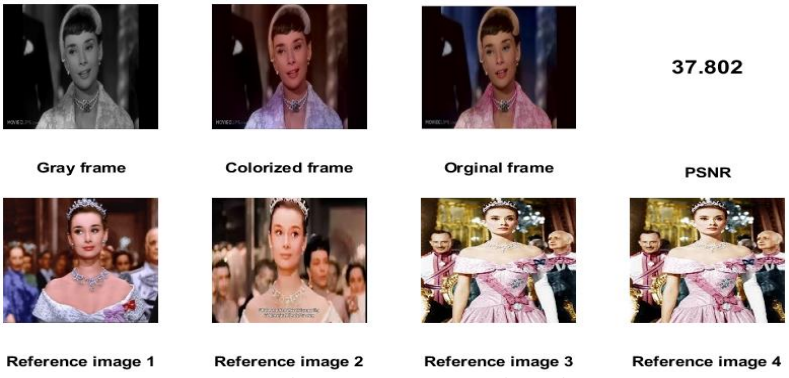

Fig. 8. (d) Grayscale frame colorization using four reference images

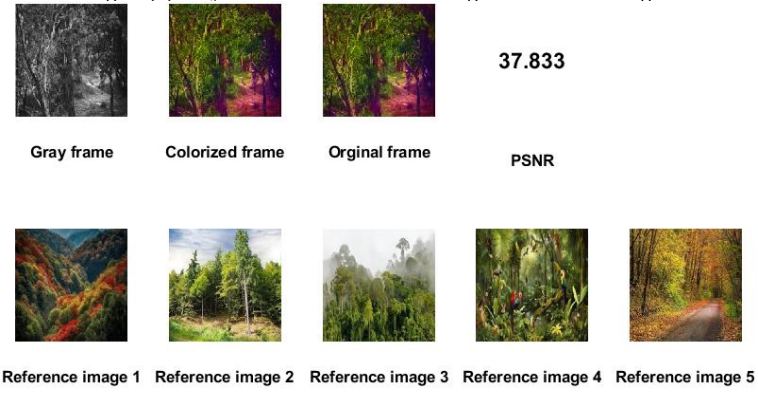

Fig. 8. (e) Grayscale frame colorization using five reference images

Fig. 8. PSNR Values for Videos Colorized via the Proposed Method with One, Two, Three, Four, and Five Reference Images. Subfigures (a) through (e) Demonstrate greyscale frame colorization via five reference images.

The proposed algorithm was then applied, and various grayscale videos were used to determine its accuracy in terms of quantitative measurements and objective visual comparisons. The PSNR was used as an important parameter to evaluate the accuracy of the colorization step performed by the script we designed. For reliability, the evaluation used grayscale videos obtained from the colored videos and allowed for side-by-side comparisons with the colorization results and the ground truth videos. This approach improved the reliability of the PSNR analysis because Fig. 8 shows that the colorized videos are very similar to the ground truth, confirming the success of the proposed method. One of the main features affected by the choice of reference images was the quality of the colorized images.

The quality of the colorized videos was assessed via various metrics, as summarized in Table I. The results indicate predominantly positive performance, with most metrics demonstrating a strong resemblance between the colorized videos and their ground truths. Achieving accurate colorization in a fully automated manner, especially without prior knowledge of the original colors, poses significant challenges. However, the metrics in Table I underscore the effectiveness and reliability of the proposed method. These results validate its ability to replicate the true colors of the original video, even in a blind colorization scenario, showcasing its potential for high-quality automated video colorization.

TABLE I. THE MEASURE OF VIDEO QUALITY AFTER THE COLORIZATION PROCEDURE.

Number of references	AD	RMSE	MD	SC	NAE	NCC	SSIM	PCC
1	0.45	4.443	1.362	0.945	0.893	0.887	0.980	0.984
2	0.31	3.719	0.481	0.902	0.732	0.896	0.987	0.989
3	0.25	3.372	0.254	0.821	0.624	0.99	0.993	0.993
4	0.22	3.340	0.131	0.758	0.518	0.993	0.996	0.995
5	0.14	3.328	0.138	0.673	0.357	0.998	0.998	0.997

We conducted a detailed visual comparison of our results with those illustrated in Fig. 9, which showcases the outputs of other studies. Our method demonstrated superior performance, with colorized videos exhibiting more realistic and consistent colorization than existing techniques do. This visual comparison underscores the effectiveness of our proposed algorithm in achieving better colorization quality. The improved results validate the robustness of our approach in addressing the challenges of video colorization.

Fig. 9. Visual comparison of the colorization process for various methods.

The superior quality of the proposed method is further corroborated by the PSNR and SSIM metrics summarized in Table II. These metrics quantify the difference between the colorized videos and their ground truths, indicating how closely the recovered videos align with the original-colored videos. The results clearly demonstrate that our proposed method achieves the highest SSIM value of 0.999982 and PSNR of 37.89, surpassing all other existing techniques. For example, studies [22] and [26] reported SSIM values of 0.963565 and 0.98229 and PSNR values of 34.57 and 35.56, which are significantly lower than those achieved by our approach.

As detailed in the analysis, our method excels in maintaining structural consistency and enhancing perceptual appearance. While the closest competitor attained an SSIM of 0.999959 and a PSNR of 36.98 [29], our method provides a slight yet significant improvement in accurately representing the true colors of videos. These advancements affirm the ability of our approach to address limitations in prior methods, such as suboptimal color quality and lack of temporal continuity.

By comparing the PSNR, SSIM values, and visual quality, we establish that the proposed algorithm outperforms state-of-the-art methods. This robust performance highlights its reliability and capability as a fully automated video colorization solution. These results lay a strong foundation for the method's real-world application in producing highly accurate and visually consistent colorizations of grayscale videos.

TABLE II. COMPARISON OF VIDEO QUALITY AFTER THE COLORIZATION PROCEDURE USING PSNR AND SSIM MEASURES

NO	Reference	SSIM	PSNR
1	[23]	0.963565	34.57
2	[24]	0.982004	35.31
3	[25]	0.96376	35.12
4	[26]	0.963755	34.81
5	[27]	0.98229	35.56
6	[28]	0.999852	36.5
7	[10]	0.987278	36.21
8	[29]	0.987129	35.8
9	[30]	0.999959	36.98
10	Our	0.999982	37.89

7. CONCLUSION

In conclusion, the proposed methodology presents a robust and efficient framework for video colorization by integrating three key phases: preprocessing, reference image processing, and source-reference network application. It also uses techniques such as multilayer feature fusion, bidirectional optical flow, and attention-based source–reference color matching to ensure color accuracy as well as temporal continuity over frames in the video sequence. This trade-off between color accuracy and temporal stability greatly improves the quality of colorized videos. However, there are some weaknesses associated with this approach, the principal of which is the necessity of using reference images; inadequate or low-quality references might result in erroneous colorization. However, the method has strengths in solving two important problems related to video colorization: high color accuracy and temporal consistency. Some of the main contributions of this work are as follows: Compared with existing methods, enhanced attention mechanisms and source-reference matching substantially enhance performance. The results of this research will help to expand the development of more accurate and faster video colorization techniques for different tasks.

Conflicts of interest

The authors declare that they have no conflicts of interest.

Funding

The acknowledgements section of the paper does not mention any financial support from institutions or sponsors.

Acknowledgement

We would like to express our sincere gratitude to everyone who provided invaluable guidance and insightful feedback throughout this research. The expertise and advice we received were crucial in shaping the direction of our work and ensuring its successful completion.

References

[1] T. Wijesinghe, C. Abeysinghe, C. Wijayakoon, L. Jayathilake, and U. Thayasivam, "FlowChroma - A Deep Recurrent Neural Network for Video Colorization," in Image Analysis and Recognition. ICIAR 2020. Lecture Notes in Computer Science, vol. 12131, A. Campilho, F. Karray, and Z. Wang, Eds., Cham: Springer, 2020, pp. 16–29, doi: 10.1007/978-3-030-50347-5_2.

[2] S. Huang, X. Jin, Q. Jiang, and L. Liu, "Deep Learning for Image Colorization: Current and Future Prospects," Engineering Applications of Artificial Intelligence, vol. 114, pp. 105006, 2022.

[3] C. Lei and Q. Chen, "Fully Automatic Video Colorization with Self-Regularization and Diversity," in Proc. IEEE/CVF Conf. Computer Vision and Pattern Recognition, 2019, pp. 3753–3761.

[4] E. Saleem and N. K. El Abbadi, "Auto Colorization of A Gray-Scale Image Using YCbCr Color Space," Iraqi J. Sci., vol. 61, no. 12, pp. 3379–3386, 2020.

[5] M. E. A-Monem and T. Z. Hammood, "Video Colorization Methods: a Survey," Iraqi J. Sci., vol. 61, no. 3, pp. 675–686, 2020.

[6] A. Alsajri, H. A. Salman, and A. Steiti, “Generative Models in Natural Language Processing: A Comparative Study of ChatGPT and Gemini”, Babylonian Journal of Artificial Intelligence, vol. 2024, pp. 134–145, Nov. 2024.

[7] K. Tejashwini, M. Soumya, P. Kalyani, G. Deepika, and C. Sujatha, "Colorization of Black and White Images and Video Using CNN," in Advances in Computational Intelligence and Its Applications, CRC Press, 2024, pp. 296–306.

[8] G. Zhang, X. Hong, Y. Liu, Y. Qian, and X. Cai, "Video Colorization Based on Variational Autoencoder," Electronics, vol. 13, no. 12, 2024.

[9] Y. Yang, J. Jin, Y. Huang, K. Guo, and X. Xu, "Reference-based Video Colorization with AB Chrominance Point and Temporal Propagation," in Proc. 16th Int. Conf. Machine Learning and Computing (ICMLC '24), 2024, pp. 340–346.

[10] S. Chen et al., "Exemplar-based video colorization with long-term spatiotemporal dependency," Knowledge-Based Systems, vol. 284, pp. 111240, 2024.

[11] O. M. Hammad, I. Smaoui, A. Fakhfakh, and M. M. Hashim, “Recent advances in digital image masking techniques Future challenges and trends: a review”, SHIFRA, vol. 2024, pp. 67–73, May 2024, doi: 10.70470/SHIFRA/2024/008.

[12] F. Dogan and I. Turkoglu, "Comparison of Deep Learning Models in Terms of Multiple Object Detection on Satellite Images," J. Eng. Res., vol. 10, no. 3A, 2022.

[13] S. Park and Y. G. Shin, "Generative residual block for image generation," Applied Intelligence, vol. 52, pp. 7808–7817, 2022, doi: 10.1007/s10489-021-02858-6.

[14] V. Thakkar et al., "Batch Normalization in Convolutional Neural Networks — A comparative study with CIFAR-10 data," in 2018 5th Int. Conf. Emerging Applications of Information Technology (EAIT), 2018, pp. 1–5.

[15] A. K. Dubey and V. Jain, "Comparative Study of Convolution Neural Network’s Relu and Leaky-Relu Activation Functions," in Applications of Computing, Automation and Wireless Systems in Electrical Engineering. Lecture Notes in Electrical Engineering, vol. 553, S. Mishra, Y. Sood, and A. Tomar, Eds., Springer, 2019, doi: 10.1007/978-981-13-6772-4_76.

[16] C. Ma, X. Mu, and D. Sha, "Multilayers Feature Fusion of Convolutional Neural Network for the Scene Classification of Remote Sensing," IEEE Access, vol. 7, pp. 121685–121694, 2019.

[17] D. A. Clevert, T. Unterthiner, and S. Hochreiter, "Fast and Accurate Deep Network Learning by Exponential Linear Units (ELUs)," arXiv preprint, arXiv:1511.07289, 2016.

[18] D. Wang, J. Wang, W. Li, and P. Guan, "T-CNN: Trilinear Convolutional Neural Networks Model for Visual Detection of Plant Diseases," Comput. Electron. Agric., vol. 190, pp. 106468, 2021.

[19] A. Radford et al., "Learning Transferable Visual Models from Natural Language Supervision," in Proc. Int. Conf. Machine Learning, PMLR, 2021, pp. 8748–8763.

[20] H. Tan and M. Bansal, "Lxmert: Learning Cross-Modality Encoder Representations from Transformers," arXiv preprint, arXiv:1908.07490, 2019.

[21] N. K. El Abbadi and E. Saleem, "Gray Image Colorization Based on General Singular Value Decomposition and YCbCr Color Space," Kuwait J. Sci., vol. 46, no. 4, 2019.

[22] A. S. Ketab and N. K. El Abbadi, "LipPrint: Using Lip Movements as a Silent Password," Mesopotamian J. Cybersecurity, vol. 4, no. 2, pp. 74–87, 2024.

[23] B. Xie, S. Yang, Y. Li, Y. Xu, and G. Wang, "Color Transfer Based on Cartoon Decomposition and Efficient Illuminant Estimation," in Int. Conf. Signal Processing and Communication Technology (SPCT 2022), SPIE, 2023, p. 1261502.

[24] Z. Sheng, H.-L. Shen, B. Yao, and H. Zhang, "Guided Colorization Using Mono-Color Image Pairs," IEEE Trans. Image Process., vol. 32, pp. 905–920, 2023.

[25] Z. Ke, Y. Liu, L. Zhu, N. Zhao, and R. W. H. Lau, "Neural Preset for Color Style Transfer," in Proc. IEEE/CVF Conf. Computer Vision and Pattern Recognition, 2023, pp. 14173–14182.

[26] V. U and G. V, "Superpixel based Image Colorization with Automated Reference Image Selection," in 2023 IEEE Int. Students' Conf. Electrical, Electronics and Computer Science (SCEECS), Bhopal, India, 2023, pp. 1–6, doi: 10.1109/SCEECS57921.2023.10061822.

[27] X. Zhu and E. Hu, "A Local Color Transfer Method Based on Optimal Transmission," in 2023 5th Int. Conf. Artificial Intelligence and Computer Science (AICS 2023), 2023, pp. 483–492.

[28] C. Lv, D. Zhang, S. Geng, Z. Wu, and H. Huang, "Color Transfer for Images: A Survey," ACM Trans. Multimedia Comput. Commun. Appl., vol. 20, 2023, doi: 10.1145/3635152.

[29] C. Lv and D. Zhang, "Palette-based Color Transfer between Images," arXiv preprint, arXiv:2405.08263, 2024.

[30] S. Bao, Y. Zhao, Y. Ji, et al., "Color Transfer Method Based on Saliency Features for Color Images," *Opt. Rev.*, vol. 31, pp. 369–382, 2024, doi: 10.1007/s10043-024-00888-2.