

An efficient direction oriented block-based video inpainting using morphological operations and adaptively dimensioned search region with direction-oriented block-based inpainting

Shyni Shajahan¹, Y. Jacob Vetha Raj²

¹Department of Computer Science and Engineering, Adi Shankara Institute of Engineering and Technology, Kalady, India

²Department of Computer Science and Engineering, Nesamony Memorial Christian College, Tamilnadu, India

Article Info

Article history:

Received Jul 25, 2024

Revised Jul 3, 2025

Accepted Jul 12, 2025

Keywords:

Bonomially distributed foreground segmentation network
Adaptively dimensioned search region with direction oriented block based inpainting
Morphological operations
Sum of squared differences
Video inpainting

ABSTRACT

Video inpainting is a technique in computer vision used to remove unwanted objects from video sequences while preserving visual consistency, so that modifications remain unnoticeable to the human eye. This paper presents an accurate video inpainting model based on the adaptively dimensioned search region with direction-oriented block-based inpainting (ADSR-DOBI) algorithm. The model operates in five main phases: preprocessing, background separation, morphological operations, object removal, and video inpainting. Initially, the input video is converted into frames, followed by preprocessing steps such as deionizing and resizing. These frames are then processed using a background subtraction module, where object localization and foreground detection are performed using the binomially distributed foreground segmentation network (BDFgSegNet) and morphological techniques. This results in segmented foreground objects tracked across frames. The object removal phase eliminates the identified foreground objects and defines the missing regions (holes) to be filled. The ADSR-DOBI algorithm is then applied to inpaint these regions seamlessly. Experimental results demonstrate that this approach outperforms existing state-of-the-art methods in both accuracy and efficiency.

This is an open access article under the [CC BY-SA](https://creativecommons.org/licenses/by-sa/4.0/) license.



Corresponding Author:

Shyni Shajahan

Department of Computer Science and Engineering, Adi Shankara Institute of Engineering and Technology
Kalady, Kerala, India

Email: shyni.cs@adishankara.ac.in

1. INTRODUCTION

Inpainting involves filling missing pixels in images with visually plausible content [1], but it is inherently ill-posed with no unique solution [2]. The need for inpainting has increased with the growth of high-resolution multimedia content [3]. Video inpainting extends this task to temporal data, aiming to fill missing regions across frames with coherent and realistic content [4], [5]. Challenges arise from camera motion and complex object movements in real-world videos [6]. Applying image inpainting models frame-by-frame often leads to temporal inconsistencies like flickering [7]. This naive approach overlooks video dynamics and fails to capture motion-driven appearance changes over time [8], highlighting the importance of spatiotemporal coherence in high-quality video inpainting.

As videos exhibit temporal regularity, to inpaint a given frame, it is natural to use data from other frames as the data in other frames may correspond to parts of the scene behind the masked region [9]. Both spatial structure and temporal coherence are required to be considered in high-quality video inpainting [10]. Recovering missing video content requires the understanding of not only the spatial context of each frame but

also the motion context across frames [11]. For any missing pixels that lack good correspondence due to occlusion, the video inpainting method must hallucinate reasonable content [12]. The state-of-the-art methods tend to capture long-term correspondences with an attention mechanism, so the available content at distant frames can be globally propagated to the unknown regions [13]. Traditional patch-based methods find similar Spatio-temporal patches from the known regions of videos to fill the holes, which formulate the problem as a patch-based optimization task [14]. These methods rely heavily on the hypothesis that the missing content in the corrupted region appears in neighboring frames, which greatly limits their generalization ability [15]. In addition, patch-based methods assume there is a reference for the missing part and often fail to recover non-repetitive and complex regions (e.g. they cannot recover a missing face well) [16]. In recent years, a number of deep learning-based video inpainting methods are proposed [17]. These exiting deep video inpainting methods can be summarized as two key modules, a temporal feature aggregation, and single-frame inpainting for temporal consistency [18].

Siddavatam *et al.* [19] proposed a video inpainting method using autoencoders that learns the background first, then object features, followed by object removal and background reconstruction. They used a pre-trained YOLO model for object detection. Although the method showed improved performance, it faced limitations related to deepfake tasks. Ke *et al.* [20] introduced an occlusion-aware video object inpainting approach with the YouTube-VOI benchmark for realistic occlusions. Their video object inpainting network (VOIN) used temporal GANs and spatio-temporal attention for shape completion and texture generation. While effective for complex objects, VOIN's performance degraded with inaccurate input. Szeto *et al.* [21] proposed a temporally-aware interpolation network for video frame inpainting, using a video prediction subnetwork to generate intermediate frames and blending them with temporally-aware interpolation (TAI). Their method outperformed state-of-the-art approaches but produced blurry results under heavy camera motion. Huang and Lin [22] introduced a video inpainting method based on object motion rate and color variance, using an adaptive foreground model and exemplar-based inpainting for unpaired areas. While their approach yielded visually pleasing results, it struggled to accurately estimate motion rates when moving objects overlapped. Inpainted videos have become more and more difficult to be distinguished even by eyes in pace with the remarkable success in video inpainting methods [23]. The difficulty of video inpainting is inherently tied to the content of the videos and masks being inpainted. So, content-informed diagnostic evaluation is performed, which identifies the strengths and weaknesses of modern inpainting methods [24]. Most of the existing techniques developed for video inpainting have complexities in terms of computation and accuracy. Although there are several techniques, there is a constant demand for reliable and efficient video inpainting systems. Therefore, this paper proposes an efficient direction oriented fast iterative block-based video inpainting model using ADSR-DOBI.

The rest of the paper is organized as follows, section 1 surveys the existing works related to video inpainting. Section 2 explains the proposed methodology. The experimental evaluation of the proposed methodology is given in section 3 and section 4 concludes the paper with future enhancement.

2. PROPOSED VIDEO INPAINTING SYSTEM

In this paper, direction oriented fast iterative block-based video inpainting using morphological operations and SSD is done. The proposed method first detects the foreground object that needs to be removed and the target region to be inpainted. Then the ADSR-DOBI algorithm is utilized for the purpose of inpainting, where the target region is inpainted with the efficient block matching mechanism. The block diagram of the proposed methodology is shown in Figure 1.

2.1. Processing

In this proposed methodology, the input video (V_{in}) is taken from publicly available datasets. As part of pre-processing, frames are extracted from the captured videos for the further process and the converted frames are initialized as,

$$fr_{(n)} = V_{in}\{fr_{(1)}, fr_{(2)}, \dots, fr_{(N)}\} \quad (1)$$

where, $fr_{(n)}$ is the number of frames. The converted frames are further pre-processed based on the following steps.

- a. Resize image: In this section, the frames $fr_{(n)}$ are resized for reducing the computational time of the system. Image resizing refers to the scaling of images. It helps in reducing the number of pixels from an image and also zooming in on images. Because the large images are fed into the AI algorithm vary in size therefore the training might be increased.

- b. Noise Removal: In this section, the noises are removed due to the presence of blur and illuminations in the images. Hence the proposed method uses the technique of Gaussian smoothing in order to enhance the image structures at a different scale. The visual effect of this blurring technique is a smooth blur resembling that of viewing the image through a translucent screen. The degree of smoothing is determined by the standard deviation of the Gaussian.

Hence the preprocessed frame is initialized as,

$$fr_{pp(n)} = \{fr_{pp(1)}, fr_{pp(2)}, \dots, fr_{pp(N)}\} \quad (2)$$

where, $fr_{pp(n)}$ denotes the preprocessed frames.

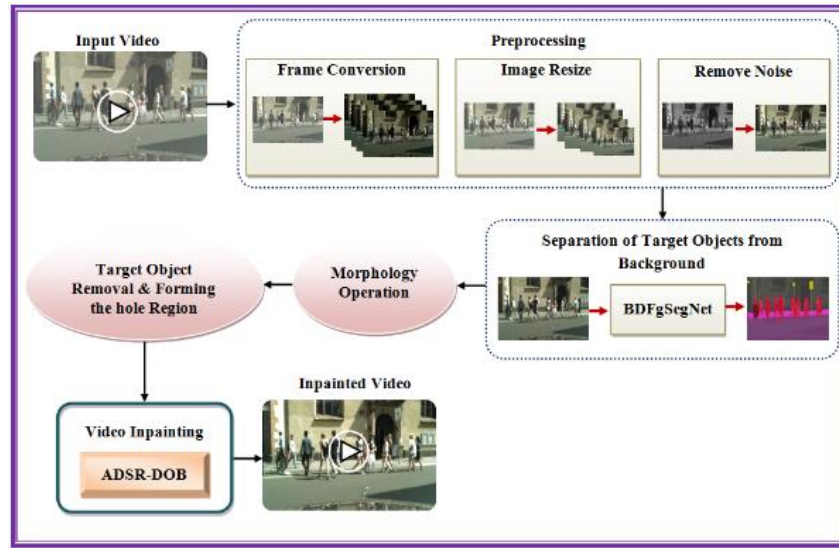


Figure 1. Block diagram of the proposed methodology

2.2. Background subtraction

Background subtraction is a widely used method for detecting moving objects in videos captured by static cameras. It helps in segmenting foreground blobs into individual objects and tracking them across frames. In this work, foreground segmentation network (FgSegNet) is used for this purpose. FgSegNet is a recently developed, high-performing neural network that employs encoder-decoder architecture. The encoder, made up of convolutional neural networks (CNNs), extracts image features, while the decoder, using a transposed CNN (TCNN), reconstructs the feature maps for object segmentation. This architecture enables accurate background subtraction and object identification. To ensure stable training and avoid issues like vanishing or exploding gradients, the network's weights are initialized using a Binomial distribution. The architecture of FgSegNet is shown in Figure 2.

Initially, the input image frames are fed into three CNNs and the outputs are concatenated and applied for TCNN. Before segmentation, the weights of networks must be initialized. Here the weights are initialized using the Binomial distribution function rather than assigning the random numbers. Thus, the weights are initialized as,

$$Z_E = \binom{m}{E} G^E J^{m-E} \quad (3)$$

where, Z_E denotes the binomial probability, G denotes the probability of success, J probability of failure, m denotes number of trials, E denotes the specific outcomes *i.e.*, weights.

2.2.1. Encoder network

The encoder network consists of three copies of CNN, each of which contains the first four blocks as that of the VGG-16 net and the dropout layers. The input image frames are fed to each CNN where the convolution layer transforms the inputs into the feature maps of size $w \times h$. The transformation of feature maps can be expressed as,

$$W_{conv(n)} = \mathfrak{I}[E_n * fr_{pp(n)} + \vartheta] \quad (4)$$

$$\mathfrak{I} = \max(o, h) \text{ Where, } h = E_n * fr_{pp(n)} + \vartheta \quad (5)$$

where, $W_{conv(n)}$ denotes the output feature maps of convolution, $\mathfrak{I}$ denotes the ReLU activation function, E_n is the weight values, ϑ denotes the bias of the network. Thus, the extracted features maps are down-sampled by using the pooling layer. The pooling layer utilizes the max pooling operation that stores only the max-pooling indices i.e. the locations of maximum feature value in each pooling window to capture and store boundary information. The output of the pooling layer is computed as,

$$W_{pool(n)} = \frac{W_{conv(n)} - E_n}{s} + 1 \quad (6)$$

where, $W_{pool(n)}$ denotes the output feature maps of the pooling layer, s denotes the strides of the kernel. At the end of this process, the dropout layer is utilized to avoid the problem of overfitting.

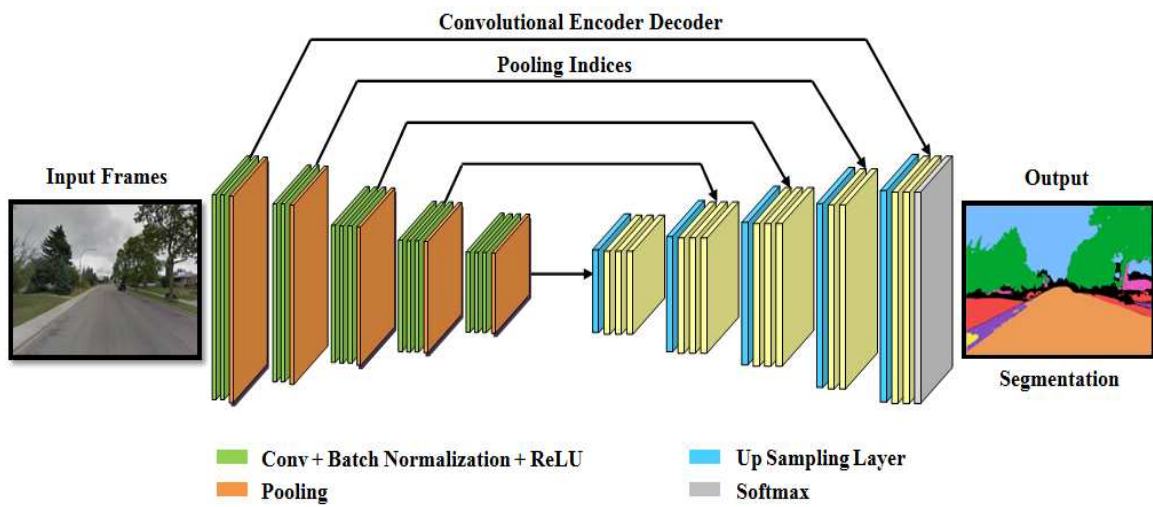


Figure 2. Architecture of FgSegNet

2.2.2. Decoder network

The output of the encoder network is concatenated to form the feature maps of different scales and then the maps are fed to TCNN for decoding the feature maps. Here the feature maps are operated with the transposed convolution to enlarge the feature maps. Finally, a sigmoid function is used in the last layer as the probability values for each pixel to obtain discrete binary class labels foreground and background. The segmented image can be obtained as,

$$fr_{seg(n)} = \frac{1}{1 + \exp(inp)} \quad (7)$$

The cross-entropy is utilized as a loss function, which is expressed as,

$$L = \frac{1}{k} \sum_{i=1}^k -(fr_{gt(i)} \log fr_{seg(i)} + (1 - fr_{gt(i)}) \log(1 - fr_{seg(i)})) \quad (8)$$

where, L denotes the loss function, k denotes the number of pixels in the frame, $fr_{gt(i)}$ is the ground truth label, $fr_{seg(i)}$ denotes the segmented labels by the network. The segmented frames are further enhanced to remove the imperfections in segmentation using morphological operations.

2.2.3. Object removal

The segmented objects or target objects specified by the morphological process $fr_{mor(n)}$ are removed from each image by forming the hole region to be inpainted. The static portion of the hole can be

filled by available background information using the video inpainting algorithm. Otherwise, image inpainting is performed based on the surrounding image statistics. Hence, the frame with the hole region is initialized as,

$$fr_{hl(n)} = [fr_{hl(1)}, fr_{hl(2)}, fr_{hl(3)}, \dots, fr_{hl(N)}] \quad (9)$$

where $fr_{hl(n)}$ denotes the frames with formed hole region, $fr_{hl(N)}$ denotes the N^{th} frame.

2.2.4. Video inpainting

This section explains the video inpainting process using the direction-oriented block-based inpainting (DOBI) algorithm. The target region, identified as the hole area, is filled with matching background content. Initially, the boundary points of the target region are determined to search for suitable patches. While using a fixed patch size increases search points, adapting the search region size improves efficiency without sacrificing quality. When motion varies among adjacent blocks, a larger search area is needed. Thus, an adaptively dimensioned search region is used, leading to the proposed adaptively dimensioned search region-based DOBI (ADSR-DOBI) method, as shown in Figure 3.

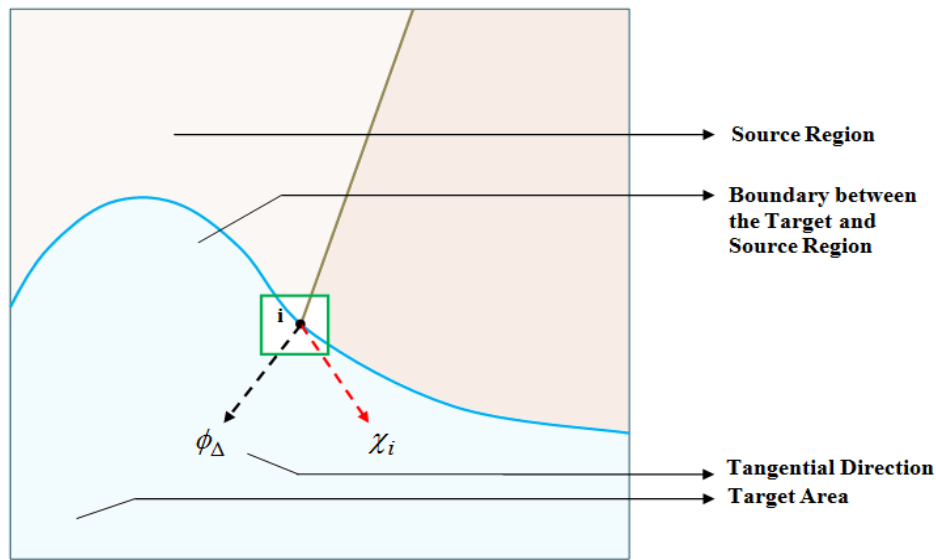


Figure 3. Frames with target and source region for inpainting

Initially, the target region (T_{TR}) has been selected and the nearest boundary points were detected. From the outside of the detected boundary, the block with the highest matching probability is selected. During this process, the dimension of the search region is chosen adaptively as,

$${}^d B_{SR} = \{B_u^d, B_v^d\} \quad (10)$$

$$B_u^d = \min\{B_u, \max(|u_n - u_1|, |u_n - u_2|, |u_n - u_3|)\} \quad (11)$$

$$B_v^d = \min\{B_v, \max(|v_n - v_1|, |v_n - v_2|, |v_n - v_3|)\} \quad (12)$$

where, (${}^d B_{SR}$) denotes the dimension of search region, (B_u^d), (B_v^d) are the adaptive displacement in the horizontal (u) and vertical (v) direction, (u_n, v_n) denotes the number of search points. The adaptive SR is bounded such that,

$$B_u^d \leq B_u, B_v^d \leq B_v \quad (13)$$

Moreover, in order to determine the pixel, the most accurate to be repaired the confidence of the repaired pixels needs to be updated. An increase in both the confidence of the neighboring pixels and the priority of the neighboring pixels constitutes the most accurate pixel to be repaired so that the improved output can be retrieved.

Therefore, the priority of a given block can be defined as,

$$\eta_i = \varpi_i \cdot \varsigma_i \quad (14)$$

$$\varpi_i = \frac{\sum_{j \in T_{TR} \cup B_{SR}} \varpi_j}{|T_{TR}|}, 0 \leq \varpi_i \leq 1 \quad (15)$$

$$\varsigma_i = \frac{|\varphi_{\Delta} \cdot \chi_i|}{\gamma}, 0 \leq \varsigma_i \leq 1 \quad (16)$$

where, η_i denotes the priority, ϖ_i, ς_i are the confidence term and data term, φ_{Δ} is the unit vector orthogonal to image gradient, χ_i is the unit vector orthogonal to the point i , γ is the normalization vector. Thus, the pixel i with the highest priority is treated as the initial search centre to choose the target patch to be filled. Then the selection of the best matching block (M_{mr}) is done based on the sum of squared differences (SSD) calculated between the known pixels of the target region and search region. From this step, the area has been identified that satisfies the following criterion as,

$$M_{mr} = \underset{T_{TR} \in B_{SR}}{\operatorname{argminr}}(M_{mr}, T_{TR}) \quad (17)$$

where, $r(\bullet)$ is the SSD between the block (M_{mr}) and the block (T_{TR}). Hence the corresponding position to the unknown pixel i of the target region is filled by assigning the known pixels of the matching block j obtained. After filling the confidence value is updated as,

$$\eta_j = \eta_i, \forall j \in M_{mr} \cap T_{TR} \quad (18)$$

The steps are repeated until the target region gets filled. The pseudo-code of the proposed ADSR-DOBI is shown in below Figure 4.

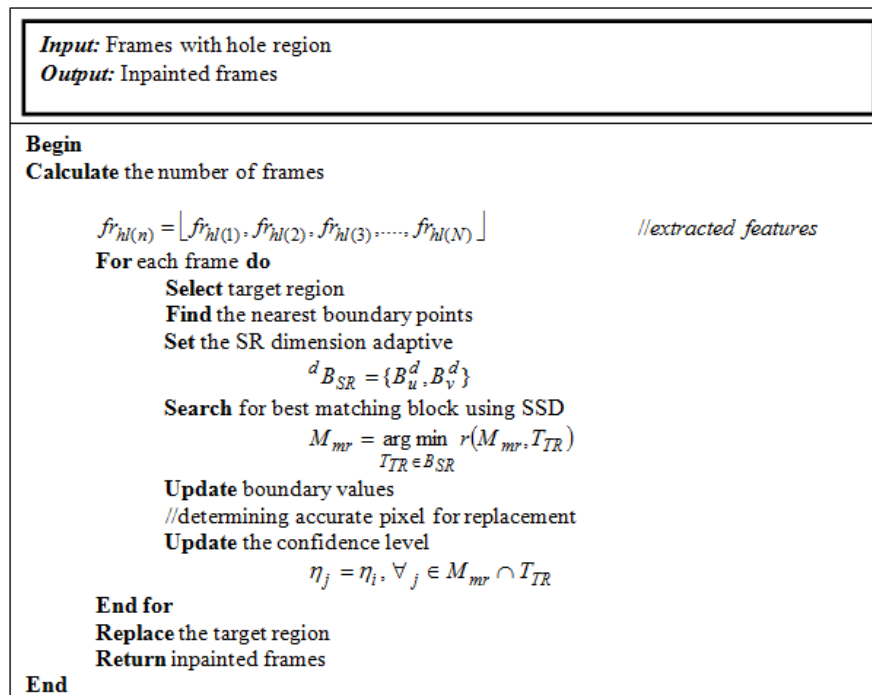


Figure 4. Pseudo code of the ADSR-DOBI algorithm

3. RESULTS AND DISCUSSION

In this section, the performance of the proposed video inpainting method is analyzed. The proposed methodology is implemented in the working platform of PYTHON.

3.1. Database description

For the performance analysis, the proposed work uses the YouTube-video object segmentation (VOS) dataset that is publically available on the internet. YouTube-VOS contains 4,453 videos. From the dataset, 80% of data was used for training and 20% data for testing. The collected dataset has more than 7,800 unique objects, 190k high-quality manual annotations, and more than 340 minutes in duration.

3.2. Performance analysis

The proposed BDFgSegNet method is based on quality metrics such as sensitivity, specificity, accuracy, precision, recall, F-measure, false positive rate (FPR), false negative rate (FNR), and Matthews correlation coefficient (MCC). Table 1 shows the comparative analysis of the proposed ADSR-DOBI algorithm and the existing DOS-based algorithm. The analysis has been done by varying the size of the frame. For a 28541-pixel frame, the time taken for inpainting is 0.018 seconds lesser than the existing DOS-based method. Similarly, for a 54300-pixel frame, the time taken for inpainting is 1.2267 seconds lesser when compared with the DOS algorithm. Lesser time, taken for inpainting the given area, shows the efficiency of the proposed method.

Table 2 presents a performance comparison of the proposed ADSR-DOBI method with existing techniques using quality metrics such as PSNR, SSIM, MSE, and RMSE. A higher PSNR and SSIM indicate better image quality, while lower MSE and RMSE reflect reduced error. The proposed method shows a 0.87 dB improvement in PSNR over FFBMA and a 0.07 increase in SSIM compared to BBGDS. Additionally, MSE and RMSE values are consistently lower than those of existing methods. Overall, ADSR-DOBI outperforms other techniques across all metrics, demonstrating its effectiveness in video inpainting.

Table 3 presents a performance comparison between the proposed BDFgSegNet and existing methods using various quality metrics. Higher values of accuracy, specificity, sensitivity, precision, F-measure, and MCC indicate better performance, while lower FPR and FNR values reflect greater efficiency. As observed, existing methods such as CNN, ANN, DNN, and SegNet show relatively low performance across these metrics. In contrast, the proposed BDFgSegNet demonstrates superior results in all evaluated metrics, highlighting its effectiveness with optimal feature selection.

Table 1. Performance comparison of proposed ADSR-DOBI algorithm with the DOS-based algorithm

Total frame size (in pixels)	Removed area size (in pixels)	Time taken in seconds	
		DOS-based algorithm	Proposed ADSR-DOBI
28541	296	0.15	0.132
54300	3480	1.856	0.6293
37000	5210	2.5706	0.5674
92982	6015	2.7522	0.9344
195860	25806	10.2932	3.4372

Table 2. Performance analysis of proposed ADSR-DOBI method based on different image quality metrics

Techniques	PSNR	SSIM	MSE	RMSE
DOS	14.19	0.75	21.95	4.69
BBGDS	17.16	0.78	17.29	4.16
FFBMA	21.32	0.8	15.81	3.97
Proposed ADSR-DOBI	22.19	0.85	14.22	3.77

Table 3. Performance analysis of proposed BDFgSegNet method based on quality metrics

Performance metrics/techniques	SegNet	CNN	ANN	DNN	Proposed BDFgSegNet
Accuracy	92.23	91.31	94.34	92.26	97.14
Specificity	86.56	87.27	85.92	92.39	93.65
Sensitivity	89.72	90.24	85.24	90.62	95.17
Precision	92.56	90.31	87.34	94.83	97.29
F-Measures	93.67	92.98	88.29	93.62	97.48
FPR	46.14	44.15	59.82	30.51	11.16
FNR	42.32	49.42	55.68	37.58	8.98
MCC	85.42	82.31	80.92	87.28	94.53

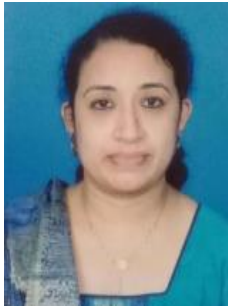
4. CONCLUSION

Video inpainting removes or restores missing regions in a video using spatial and temporal information. This paper proposes an efficient ADSR-DOBI block matching algorithm for inpainting, where the target region is identified and filled with matching background content. The proposed method is evaluated against

existing techniques using quality metrics. Results show that ADSR-DOBI achieves superior performance, with a PSNR value of 22.19 and an inpainting time of 3.43 seconds—both better than existing methods. These findings demonstrate the efficiency of ADSR-DOBI. Future work can improve the method further by addressing occluded regions and enhancing motion handling capabilities.

REFERENCES

- [1] S. Lee, S. W. Oh, D. Won, and S. J. Kim, "Copy-and-paste networks for deep video inpainting," *arXiv:1908.11587*, Aug. 2019, doi: 10.48550/arXiv.1908.11587.
- [2] H. Zhang, L. Mai, H. Jin, Z. Wang, N. Xu, and J. Collomosse, "An internal learning approach to video inpainting," in *2019 IEEE/CVF International Conference on Computer Vision (ICCV)*, Oct. 2019, pp. 2720–2729, doi: 10.1109/iccv.2019.00281.
- [3] H. Ouyang, T. Wang, and Q. Chen, "Internal video inpainting by implicit long-range propagation," in *2021 IEEE/CVF International Conference on Computer Vision (ICCV)*, Oct. 2021, pp. 14559–14568, doi: 10.1109/iccv48922.2021.01431.
- [4] R. Liu *et al.*, "FuseFormer: fusing fine-grained information in transformers for video inpainting," in *2021 IEEE/CVF International Conference on Computer Vision (ICCV)*, Oct. 2021, pp. 14020–14029, doi: 10.1109/iccv48922.2021.01378.
- [5] G. Tudavekar, S. R. Patil, and S. S. Saraf, "Dual-tree complex wavelet transform and super-resolution based video inpainting application to object removal and error concealment," *CAAI Transactions on Intelligence Technology*, vol. 5, no. 4, pp. 314–319, Nov. 2020, doi: 10.1049/trit.2019.0045.
- [6] R. Xu, X. Li, B. Zhou, and C. C. Loy, "Deep flow-guided video inpainting," in *2019 IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR)*, Jun. 2019, pp. 3718–3727, doi: 10.1109/cvpr.2019.00384.
- [7] Y.-L. Chang, Z. Y. Liu, K.-Y. Lee, and W. Hsu, "Free-form video inpainting with 3D gated convolution and temporal PatchGAN," in *2019 IEEE/CVF International Conference on Computer Vision (ICCV)*, Oct. 2019, pp. 9065–9074, doi: 10.1109/iccv.2019.00916.
- [8] D. Kim, S. Woo, J.-Y. Lee, and I. S. Kweon, "Deep video inpainting," in *2019 IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR)*, Jun. 2019, pp. 5785–5794, doi: 10.1109/cvpr.2019.00594.
- [9] D. Lao, P. Zhu, P. Wonka, and G. Sundaramoorthi, "Flow-guided video inpainting with scene templates," *2021 IEEE/CVF International Conference on Computer Vision (ICCV)*, Montreal, QC, Canada, 2021, pp. 14579–14588, doi: 10.1109/iccv48922.2021.01433.
- [10] Z. Li, C.-Z. Lu, J. Qin, C.-L. Guo, and M.-M. Cheng, "Towards an end-to-end framework for flow-guided video inpainting," *arXiv:2204.02663*, Apr. 2022, doi: 10.48550/arXiv.2204.02663.
- [11] C. Wang, H. Huang, X. Han, and J. Wang, "Video inpainting by jointly learning temporal structure and spatial details," *Proceedings of the AAAI Conference on Artificial Intelligence*, vol. 33, no. 01, pp. 5232–5239, Jul. 2019, doi: 10.1609/aaai.v33i01.33015232.
- [12] X. Zou, L. Yang, D. Liu, and Y. Jae Lee, "Progressive temporal feature alignment network for video inpainting," *2021 IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR)*. IEEE, pp. 16443–16452, Jun. 2021, doi: 10.1109/cvpr46437.2021.01618.
- [13] J. Ren, Q. Zheng, Y. Zhao, X. Xu, and C. Li, "DLFormer: discrete latent transformer for video inpainting," in *2022 IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR)*, Jun. 2022, pp. 3501–3510, doi: 10.1109/cvpr52688.2022.00350.
- [14] R. Liu, Z. Weng, Y. Zhu, and B. Li, "Temporal adaptive alignment network for deep video inpainting," in *Proceedings of the Twenty-Ninth International Joint Conference on Artificial Intelligence*, Jul. 2020, pp. 927–933, doi: 10.24963/ijcai.2020/129.
- [15] C. Wang, X. Chen, S. Min, J. Wang, and Z.-J. Zha, "Structure-guided deep video inpainting," *IEEE Transactions on Circuits and Systems for Video Technology*, vol. 31, no. 8, pp. 2953–2965, Aug. 2021, doi: 10.1109/tcsvt.2020.3034422.
- [16] Y.-L. Chang, Z. Y. Liu, and W. Hsu, "VORNet: spatio-temporally consistent video inpainting for object removal," in *2019 IEEE/CVF Conference on Computer Vision and Pattern Recognition Workshops (CVPRW)*, Jun. 2019, pp. 1785–1794, doi: 10.1109/cvprw.2019.00229.
- [17] K. Zhang, J. Fu, and D. Liu, "Inertia-guided flow completion and style fusion for video inpainting," in *2022 IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR)*, Jun. 2022, pp. 5972–5981, doi: 10.1109/cvpr52688.2022.00589.
- [18] X. Ding, Y. Pan, K. Luo, Y. Huang, J. Ouyang, and G. Yang, "Localization of deep video inpainting based on spatiotemporal convolution and refinement network," in *2021 IEEE International Symposium on Circuits and Systems (ISCAS)*, May 2021, pp. 1–5, doi: 10.1109/iscas51556.2021.9401675.
- [19] I. Siddavatam, A. Dalvi, D. Pawade, A. Bhatt, J. Vartak, and A. Gupta, "A novel approach for video inpainting using autoencoders," *International Journal of Information Engineering and Electronic Business*, vol. 13, no. 6, pp. 48–61, Dec. 2021, doi: 10.5815/ijieeb.2021.06.05.
- [20] L. Ke, Y.-W. Tai, and C.-K. Tang, "Occlusion-aware video object inpainting," in *2021 IEEE/CVF International Conference on Computer Vision (ICCV)*, Oct. 2021, pp. 14448–14458, doi: 10.1109/ICCV48922.2021.01420.
- [21] R. Szeto, X. Sun, K. Lu, and J. J. Corso, "A temporally aware interpolation network for video frame inpainting," *IEEE Transactions on Pattern Analysis and Machine Intelligence*, vol. 42, no. 5, pp. 1053–1068, May 2020, doi: 10.1109/TPAMI.2019.2951667.
- [22] H.-Y. Huang and C.-H. Lin, "Video inpainting using object motion rate and color variance in spatiotemporal domain," *Journal of Intelligent & Fuzzy Systems*, vol. 40, no. 3, pp. 5609–5622, Mar. 2021, doi: 10.3233/JIFS-200542.
- [23] B. Yu, W. Li, X. Li, J. Lu, and J. Zhou, "Frequency aware spatiotemporal transformers for video inpainting detection," in *2021 IEEE/CVF International Conference on Computer Vision (ICCV)*, Oct. 2021, pp. 8168–8177, doi: 10.1109/ICCV48922.2021.00808.
- [24] R. Szeto and J. J. Corso, "The devil is in the details a diagnostic evaluation benchmark for video inpainting," *arXiv:2105.05332*, 2022, doi: 10.48550/arXiv.2105.05332.

BIOGRAPHIES OF AUTHORS

Shyni Shajahan    holds a B.Tech. and M.Tech. in computer science from CUSAT and a Ph.D. from Noorul Islam Centre for Higher Education (2023). She is an assistant professor in the Department of Computer Science at Adi Shankara Institute of Engineering and Technology, Kalady. Her research interests include image processing, coding techniques, DBMS, data mining, machine learning, data science, AI, and multimedia. She has published extensively in national and international journals. She can be contacted at email: shyni.cs@adishankara.ac.in.



Y. Jacob Vetha Raj    is an associate professor in the Department of Computer Science at Nesamony Memorial Christian College, India. He received a B.E. degree in computer science from National Engineering College, Kovilpatti, India, and also received M.Tech. and Ph.D. degrees from M.S. University, Tirunelveli, India. His area of interest is image processing. He has developed many application software programs and has published numerous national and international research papers. He can be contacted at email: jacobvetharaj@gmail.com