The Visual Computer (2022) 38:295–319
https://doi.org/10.1007/s00371-020-02016-y

SURVEY

A comprehensive survey on video frame interpolation techniques

Anil Singh Parihar[1] · Disha Varshney[1] · Kshitija Pandya[1] · Ashray Aggarwal[1]

Accepted: 4 November 2020 / Published online: 4 January 2021

Abstract
Video frame interpolation is an important area in the computer vision research activities for video post-processing, surveillance, and video restoration tasks. It aims toward increasing the frame rate of a video sequence by calculating intermittent frames between consecutive input frames. This ensures extra smooth, clear motion in order to make animation fluid enough and reduce display motion blur. Advanced deep learning algorithms have the potential to discover knowledge from large-scale diverse video data. These algorithms gain insights about intermediate motion and provide new opportunities to further improve video interpolation technologies. This survey demonstrates a comprehensive overview of about a good number of contributions over past decade pertinent to the latest developments in this domain. The survey paper highlights common challenges in the area of video frame interpolation based on three key aspects: high visual quality, low complexity, and high efficiency of interpolated output from regular videos with the standard frame rate. We scrutinize the architectures, workflows, performance, advantages, and disadvantages and generate a broad categorization along with an overview of experimental results of various state-of-the-art methods executed on benchmark datasets. This survey discusses applications of diverse interpolation frameworks. It provides a backbone reference that inspires future researchers to optimize current techniques on academic and industrial grounds.

Keywords Computer vision · Video frame interpolation · Video processing · Survey

1 Introduction

There are several iconic moments in our life that we certainly wish to capture in slow motion. These events are hard to perceive thoroughly by a normal human vision: the shooting of a star, a tricky dance move, diving in a swimming pool, a cricket shot, and many others. While it is conceivable to take 240 fps recordings with a standard video recording device, it is impractical to record everything at high frame rates, as it costs large memory and is power constrained for cell phones. Moreover, very often the moments we slow down are not predictable and preferred to be recorded at standard frame rates.

✉ Anil Singh Parihar
parihar.anil@gmail.com

Disha Varshney
dishavarshney9@gmail.com

Kshitija Pandya
kshitijajain99@gmail.com

Ashray Aggarwal
ashray14aggarwal@gmail.com

[1] Machine Learning Research Laboratory, Delhi Technological University, Delhi, India

It is of extraordinary interest to produce a slow-motion video from existing recordings of high quality. Video interpolation encourages smoothness in these transformed videos of higher frame rates. It has other fascinating new applications like analyzing unlabeled videos using a supervisory signal to learn optical flow [1–3]. To the best of our knowledge, this survey on video frame interpolation task is the first of its kind in the history of the deep learning research field. Apart from this claim, this survey establishes its motive through the following contributions: (1) A comprehensive evaluation of the advanced deep learning-based methods developed in the past decade, thereby assisting readers with a detailed overview of comparative performance and research results of recent state-of-the-art methods, (2) insightful analysis of system requirements, algorithm complexity, quality of results, and characterization of methods based on different schemes of image processing and frame rate conversion, emphasizing the pros and cons of these methods outlined in reviewed works, and (3) discussion of potential challenges of frame rate conversion domain for identification of shortcomings of existing techniques and aligning potential directions for future researchers.

1.1 Requirements in industry

With the advent of fast Internet, efficient storage facilities and high-proportion compression standards, for example, MPEG-1, MPEG-2, and MPEG-4, facilitate an in-depth understanding of available video information. Subsequently, automatic detection of semantically significant events for video summarization to help video consuming, processing, and indexing is highly in demand. Numerous methodologies toward automatic event-based detection and outline in sports programs together build the computer vision literature. Among all, most strategies are created for specific games, visual editing, or explicit situations, bringing about domain-specific methodologies. For instance, some of them require the events to take place under camera surveillance, and some are confined to football matches, others to baseball, soccer, or basketball. YouTubers or National Geographic experts who make those excessively motion-controlled recordings contributed quite interesting rehearsal procedures in dance, music, and other motion-based sports and various fields of arts and cinema. The AI transformation projects intend to decide viable techniques for processing motion films to produce visually pleasing intermediate animated motion frames between existing ones by methods of interpolation. The ultimate goal is to make motion much fluid and to resolve the visual ambiguities arising from inadequate details of moving objects due to low light or poor quality of captured frames. Few use cases involve coming up with a legitimate decision on the proper punishment for any inappropriate activity by carefully analyzing close inductions about the violator's intent using slow-motion frames.

1.2 Video interpolation process

The process of generating slow-motion videos deals with extracting an enormous number of frames every second. On the off chance that we do not record enough, it gets rough and unwatchable when we speed down our video except if we utilize advanced AI methodologies to envision the additional frames by utilizing deep learning algorithms to transform 30 fps video into appealing, 240 fps slow motion. The AI framework picks two unique frames and afterward generates intermediate motion by following the development of objects starting with one frame then onto the next, as illustrated in Fig. 1. It is not equivalent to really envisioning footage like a human mind does; however, it produces close to accurate results. The procedure requires refinement before it shall flourish economically; however, when improved, it could be utilized to add slow-motion impacts to smartphone recordings of daily life events.

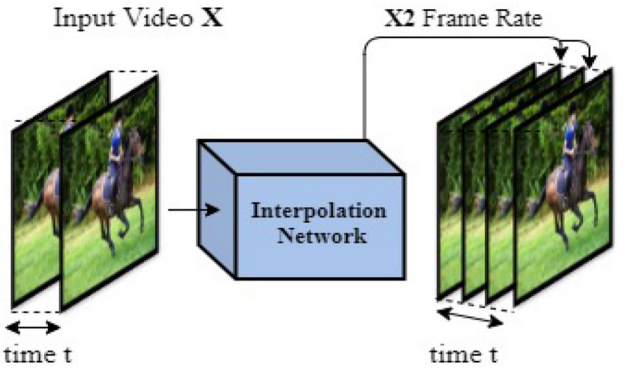


Fig. 1 General process for video frame interpolation. I_1 and I_2 are consecutive input frames fed to the interpolation network to generate intermediate frames

1.3 Drawbacks of traditional slow motion algorithms

Motion interpolation on specific brands of TVs is now and then joined by visual abnormalities in the image, as a little tear or glitch, showing up for a small amount of a second. The impact is most recognizable when the innovation all of a sudden kicks in during a quick camera pan. TV and show producers allude it as a digital artifact. The improvement of related techniques after some time has decreased the extent to which artifacts show up in modern-day commercial displays still to eliminate it overall. It is essential to study and fill in the large gap between frames. As a side effect of the apparent increment in frame rate, motion interpolation may present a "video" (versus "film") look. This look is usually alluded to as the "soap opera impact," with reference to the appearance of most communicated TV dramas or pre-2000s multi-cam sitcoms, which were typically shot utilizing a more affordable 60i video. However, the soap opera impact ruins the theatrical look of cinema works, by causing it to show up as though the watcher is either on set or viewing the background attributes. Hence, practically all makers have worked in an alternative to turn down the feature or lower the impact quality. In problem setups, generating multiple intermediate frames, the major challenge of a standard solution is not only to estimate correct motion between consecutive images but also handle occlusion to avoid severe artifacts around motion boundaries in the interpolated output as described in Fig. 8. Despite the shortcomings, different watchers still acknowledge motion interpolation, as it decreases motion ambiguities created by camera skillet and unsteady cameras, and hence yields better clarity of images. It also facilitates to build a finer frame rate of computer game programming for a progressively realistic feel, although the extra input slag might be an undesired effect. The primary contrasts between an interpolated (artificial) and commonly captured (in camera) high frame rate are that in camera is not dependent upon any

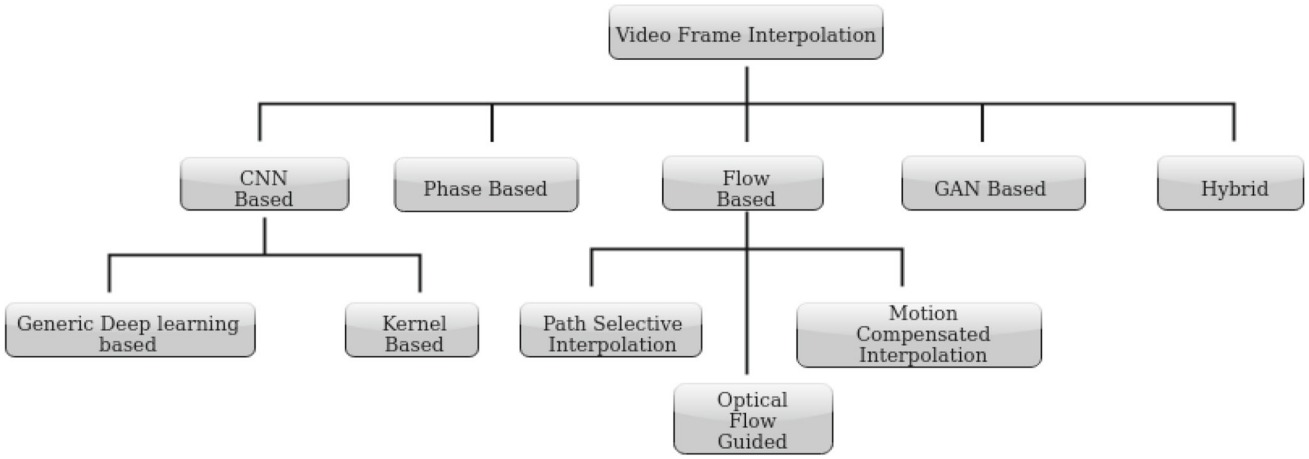


Fig. 2 Broad overview of video frame interpolation methods with technique-wise categorization designed by the author which is discussed and analyzed throughout this paper

of the previously mentioned glitches, contains progressively accurate (or "consistent with life") image data, and requires more memory and transfer speed since frames are not created in real time.

1.4 Categorical distribution

A key aspect of this survey is that it highlights five major lines of approaches to solving the aforementioned computer vision task based on the structure of network architecture and math involved named flow-based methods, CNN-based methods, phase-based methods, GAN-based methods, and hybrid methods as shown in Fig. 2. Deep learning frameworks have the edge over conventional models in terms of robustness, generalization, and learning ability. Recently, an upsurge of increasing employment of deep, fully convolutional neural networks has been eclectically explored by many researchers [1,3–5] due to their exceptional performance in solving tedious computer vision tasks.

Liu et al. [1] proposed a self-supervised framework that inherently modifies the network to compute better optical flow and warp input images to produce an intermediate frame. Considerable results are achieved in contrast to conventional supervised approaches. However, their method produces adverse artifacts such as ghosts and halo due to occlusion, which fails the optical flow estimation method. Liu et al. [6] suggest embedding a cyclic consistency loss in the training model to enhance optical flow estimation by compelling proximity between input and mapped-back images. Additionally, they also adopted edge-guided training and motion linearity to handle rich texture problems and large-scale motion despite which the method does not show any improved results for occlusion and complex motion.

Useful approaches came into existence to handle occlusion. Tianfan et al. [7] implemented a network comprising three sub-networks where the first two networks compute occlusion mask and optical flow from input frames, and the terminal network integrates interpolated frame using estimated parameters. Jiang et al. [4] handled occlusion reasoning by using visibility maps that would blend only the un-occluded pixels to the interpolated image, as shown in Fig. 7. Parallel to this cause, Bao et al. [8] exploited depth awareness, which detects occlusion explicitly by calculating additional depth maps to intermediate optical flow as described in Fig. 12. These methods show strikingly superior results than other state-of-the-art methods that perform occlusion reasoning. Nevertheless, these interpolation methods fail for high-resolution videos of beyond 4K images, which capture extensively large-scale motion. A possible solution is to improve the quality of estimated depth maps by precising color and depth consistency [9].

Niklaus et al. [10] utilized both contextual information and optical flow to design a context-aware network. Contrary to standard interpolation methods, the architecture of this network is derived from Gridnet [11], which merges warping and pixel blending into one single step. Despite achieving considerable state-of-the-art performance, this method still fails to process high-resolution video frames, due to inherent network complexity having substantial memory constraints. Meyer *et al.* [12] designed a subordinate convolution neural network less known for its texture, to handle large motion efficiently by estimating phase decomposition of the interpolated frame, as shown in Fig. 11. Niklaus et al. [5] designed a network that estimates pixel-wise spatially adaptive kernels that incorporate a mix of both pixel warping and optical flow information between consecutive input frames. Their method provides state-of-the-art performance for simple small-scale motion but demands more computational resources to process high-resolution video frames. Niklaus et al. [13] address large memory demand by constructing a method that sup-

plants 2D interpolation frame kernels with two separable 1D kernels. Since their method commands a higher computation cost than existing methods, it is incapable of processing 4K and above video frames.

To master the shortcomings of initial attempts, few methodologies are available to support high-resolution video interpolation, producing comparable results at a faster rate. Amersfoort et al. [14] proposed a remainder learning method that optimizes its performance by incorporating a multi-scale residual estimation module, which constructs the synthesized frame and anticipated flow in a coarse-to-fine trend. Peleg et al. [15] and Vidanpathirana et al. [3] proposed economic interpolated motion neural networks for HD resolution that provide real-time temporally aligned output in a block-wise manner with an impressive efficiency on standard CNN platforms [16]. Ahn et al. [17] designed a hybrid network composed of temporal and spatial interpolation sub-networks which sequentially produce a high-quality intermediate frame subject to complex structural changes and large-scale motion. Considerable state-of-the-art performance is achieved for both visual and numerical evaluations.

2 Benchmark datasets

In this section, we briefly discuss the major datasets used by authors for training their deep learning architectures for video interpolation tasks and evaluation purposes. While there are several benchmark datasets, we will be considering UCF101 [18], Middlebury [19] and Vimeo-90k [7] and some other relevant datasets like Adobe240 [20] KITTI [21] or DAVIS dataset [22] that are frequently discussed in interpolation domain. The datasets mostly contain a series of triplets, which are three consecutive frames taken from a video that acts as a single input unit. However, these datasets are somewhat used differently by different authors for training and evaluation purposes according to their learning architecture. Few experimental results are attached in Sect. 4.1 for further performance analysis.

2.1 UCF101 action recognition dataset

UCF101 [18] is an action recognition dataset collected from user-uploaded YouTube videos; hence, it contains real action sequencing videos. It has various categories for 101 different action sequences. UCF101 is an extensive dataset of the previous UCF50 dataset, which has half that is 50 categories of actions. There are 13320 videos divided into 101 categories in the UCF101 dataset, hence making it the most diverse dataset in terms of actions. Since these are user-uploaded YouTube videos, large variations in camera motion, object scale, object appearance, pose illumination conditions, cluttered background, and viewpoint are present. In total, 101 action categories are segregated into 25 groups, each having 4-7 videos of the action. Action categories are divided into human–object Interaction, body motion only, human–human interaction, playing musical instruments, and sports. Videos in the same group generally contain some common features like object appearances and background.

Most authors have not used UCF101 data for training but just for evaluation purposes, because most of the UCF101 frames only have a tiny portion of the image actually moving, while the rest is just a static background. We will present the PSNR and SSIM values for UCF101 in the results in Sect. 4. Liu et al. [1], Yu-Lun Liu et al. [6], and D. Gu et al. [23] have preferred using UCF101 for training their models. Since only a minimal area of the image is apparently moving, while the rest is just a static background, authors have selected triples with a more obvious motion by choosing those with lower PSNR values between input frames and combined UFC101 datasets with other datasets with more recognizable motion while training. All frames generally scale to the resolution of 256×256 before using them for training.

For evaluation purposes, Bao et al. for DAIN [8] and MEMC-net [24], Jiang et al. [4], Cheng et al. [25], and Yu-Lun Liu et al. [6] have used UCF101 test set, and the first and third image frames for every triplet in the dataset are used to make predictions for the second frame (temporal middle), where the resolution of images is 256×256 of pixels. While other papers use a different set of test images from the UCF101 test set, Mathieu et al. [26] used every 10th frame, i.e., 10% of UCF101 test set for evaluation and Zhang et al. [27] checked only on selected samples with apparent motion using DIS optical flow. For more recent papers like that of Ahn et al. [28], the UCF101 dataset is not useful since it majorly contains low image resolution, hence not suitable for the method which handles high-resolution video frames.

2.2 Middlebury

The Middlebury dataset [19] is an optical flow benchmark dataset that is widely used to evaluate video frame interpolation techniques. For further details on this dataset, refer to http://vision.middlebury.edu/flow/eval/. There are two subsets in Middlebury datasets. First, the *Other* set, which provides the ground-truth middle frames, second the *Evaluation* set, which hides the ground truth and is evaluated by uploading the results to the benchmark website [19]. The image resolution is 640×480 pixels in this dataset. We will compare the average interpolation error (IE) on the Middlebury dataset in the results section. Lower IE values generally indicate better performance. Using different datasets combination gives the required variability, which provides a basis for a thorough evaluation of current algorithms. There are four types of data to test different aspects of optical flow algorithms: (1) sequences with non-

rigid motion where the ground-truth flow is determined by tracking hidden fluorescent texture, (2) realistic synthetic sequences, (3) high-frame-rate video used to study interpolation error, and (4) modified stereo sequences of static scenes. Bao et al. model DAIN [8] ranks 1st in terms of normalized interpolation error and is 3rd in terms of interpolation error as on Middlebury results website http://vision.middlebury.edu/flow/eval/. The website mentioned above compares different algorithms based upon four error measures, i.e., endpoint error (EE), angular error (AE), interpolation error (IE), and normalized interpolation error (NIE).

2.3 Vimeo-90k

Vimeo-90k [7] dataset is a high-quality video clip dataset containing more than 89,000 videos of 720p or higher resolution, which are downloaded from the Vimeo video sharing platform. Authors Cheng et al. [25] and, Bao et al. for DAIN [8] and MEMC-net [24], have used Vimeo-90k dataset for training their models as it contains variable content for different scenes. The motion of objects in the Vimeo-90k dataset is much larger than that of UCF10.

Vimeo-90k dataset [7] includes 51,313 triplets for training. Each triplet is made up of 3 consecutive video frames with a resolution of 448×256 pixels. Authors have trained their networks to predict the middle frame (i.e., t = 0.5) of each triplet. There are 3,782 triplets in the test set of this dataset, and hence, it is used widely for comparing performances of different algorithms due to the high-quality videos.

2.4 Other datasets

Some other noteworthy datasets are also used for video frame interpolation. While these datasets are considerably smaller than UCF101 and Vimeo-90k, they have been successfully employed by various approaches for training, as well as testing purposes. First is Xiph[1] dataset which contains a set of 4K videos. This dataset is mainly used to test image compression. For our purposes, we either resized the 4K frames to 2K or used the centre crop of the frame to reduce the size while retaining the per-pixel motion. Next is DAVIS dataset [22] which consists of 50 high-quality videos with highly challenging frames containing occlusion and motion blur. KITTI [21] dataset has also been used for training video frame interpolation models. Most videos from which the frames are extracted are recorded from a moving platform, thus resulting in challenging scenarios. The dataset is also divided into categories which does not find a use case in frame interpolation but in robotics. Overall the size of the dataset is about 180GB. Adobe240 [20] and YouTube240 have also been employed successfully for implementing VFI models as in SuperSlomo [4].

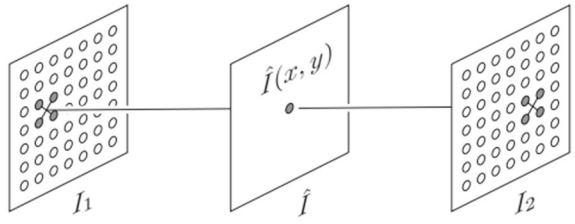
[1] https://media.xiph.org/video/derf

3 Categorized description

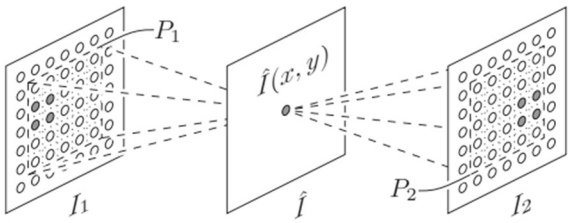
3.1 CNN- and kernel-based methods

CNN came into existence after AlexNet [29] in 2012 and has seen great popularity among data scientists. After that, it got successfully applied to many image processing applications, including optical flow estimation. CNN has been successfully applied to estimate optical flow, which can be used to generate intermediate frames. However, this two-step process results in losses of various kinds. The first attempt at generating an intermediate frame using CNN was made by Long et al. (2016) [2]. They developed a deep CNN which can be trained without any supervision. They exploited the temporal coherency that occurred naturally in the real-world videos and used it to calculate sensitivity maps, i.e., gradient with respect to input via backpropagation. Hence, it is called matching by inverting deep neural network. Its architecture is that of an auto-encoded network similar to Flownet-S [30], which shows that it incorporated the optical flow learning step within the network. The network is entirely convolutional and can be trained using any triplet of an image sequence, which can be of different resolutions. They used the Charbonneir loss function and trained it on the KITTI [21] dataset. The interpolated frames produced by the algorithms were blurry, but it opened a new avenue for data scientists working in this domain.

Niklaus et al. (2017) [5] made significant improvements to this approach. Their approach generated the pixel values of the interpolated frame by locally convolving the input frames. They proposed a fully convolutional deep neural network which predicts spatially adaptive convolutional kernels for each pixel from the two given successive input frames. They calculated a separate kernel for each pixel in the interpolated frame to estimate the value of the pixel. The predicted spatially adaptive pixel-wise convolution kernels are then convolved with the input frames to generate the interpolated frame. Traditional frame interpolation methods generally involve two steps: motion estimation and re-sampling. The convolution kernels account for both these steps. They encode both the local movement between the input frames along with the coefficient for pixel generation as described in Fig. 3. The proposed neural network can be trained end to end using widely available video data, as discussed in Sect. 2. It deals with occlusion, sudden brightness change and blur to enable high-quality video frame interpolation. Methods based on optical flow are not this flexible in handling these challenges, and these have to be dealt with



(a) Interpolation by motion estimation and color interpolation



(b) Interpolation by convolution

Fig. 3 Convolution-based approach for interpolation inspired by Niklaus et al. [5]. **(a)** A two-step general non-CNN approach following motion estimation followed by pixel synthesis based on estimated motion. Interpolation by motion estimation and color interpolation. **(b)** A direct approach for estimating convolutional kernels to convolve successive input frames for pixel color interpolation. Interpolation by convolution

separately for frame interpolation as discussed in Sect. 3.2.3. Vidanpathirana et al. [3] attempted to reduce optical flow errors by designing a pose tracking system that operates on a system of queues in a multi-threaded environment and provides a fast point tracking solution to boost the frame rate of pose estimation system.

Furthermore, much sharper results are obtained by kernel-based methods as edge-aware kernels can be estimated by the neural network. However, calculating 2D convolution kernels for each pixel is computationally expensive. Hence, the technique is comparatively slower. It also fails on videos with higher resolutions.

Niklaus et al. (2017) [13] improved on their previous approach by using separable convolutions. Instead of estimating the whole kernel for each pixel, they estimated spatially adaptive pairs of 1D convolution kernels for each pixel, thus reducing the parameters to be estimated as described in Fig. 4. It optimizes the algorithm within the allowable range. For a 1080p video frame, using separable kernels that approximate 41×41 ones only require 1.27 GB instead of 26 GB of memory. They also developed a dedicated encoder–decoder neural network to estimate kernels for all pixels in a frame at once, which gives better performance than AdaConv and more visually pleasing results. Xue et al. (2016) [31] tackled this problem in a more non-deterministic manner. The network predicts multiple extrapolated frames from a single frame. The proposed network consists of five components. First is a variational auto-encoder to encompass motion information. Second is a kernel decoder which learns motion kernels from the output of the above motion encoder. The third is an image encoder which estimated feature maps from an image. Furthermore, the next is their novel cross-convolutional layer which convolves the feature maps with motion kernels. And then finally, a regressor. The core of their network is the cross-convolution layer. It does not learn the weight of kernels but rather take feature maps and kernel weights and calculates convolution and then back propagates the gradient for both feature maps and kernels. They use conditional auto-encoder and use any simple distribution like Gaussian distribution to estimate a future frame which introduces probabilistic components in the network. Liu et al. (2017) [1] introduces a new approach in tackling this problem. It calculates dense voxel flow and uses it to generate an interpolated frame. These voxels encase the motion changes in the temporal domain, and intermediate frames can be generated using trilinear interpolation. They proposed an end to end full differentiable network that adopts a fully convolutional encoder–decoder architecture with a bottleneck layer that calculates voxel flow. This voxel flow is similar to optical flow and multiple nearest neighbor-based interpolation (MNBI) [32] but also considers time component. Since it is only an intermediate layer, it is never really evaluated. Liu et al. (2019) [6] build upon DVF and introduces a novel loss called cycle consistency loss, which can be integrated with any frame interpolation method. They postulated that given three consecutive frames I_1, I_2, and I_3, the frames generated by I_1-I_2 and I_2-I_3 would generate another frame that will be bounded by frame I_2 in a cyclic manner. This leads to better motion information preservation. CNN-based methods can only generate frames at discrete intervals of time and are generally computationally heavy.

A large proportion of existing methods tend to locate regions with relevant information to closely estimate every output pixel by applying self-produced frame warping techniques. Still, a majority of current approaches have restricted degree of freedom (DoF) and cannot fulfill real-time requirements of complex motions. To address this issue Lee et al. (2020) [33] designed the latest warping module called adaptive collaboration of flows (AdaCof) based on an operation that uses *any number of pixels* and *any location*. Unlike SepConv [13], this method calculates discrete offset vectors and kernel weights for individual target pixels to generate the output frame. It provides a more generalized warping framework in contrast to classic optical flow methods [19,34–36] and redefines majority of those as special cases of it. The network architecture comprises a fully convolutional neural network, advancing from DSepConv [37], and incorporates dual-frame adversarial loss to reasonably produce real-time intermediate frames.

However, these kernel-based interpolation techniques we discussed above fail to capture motion between frames when

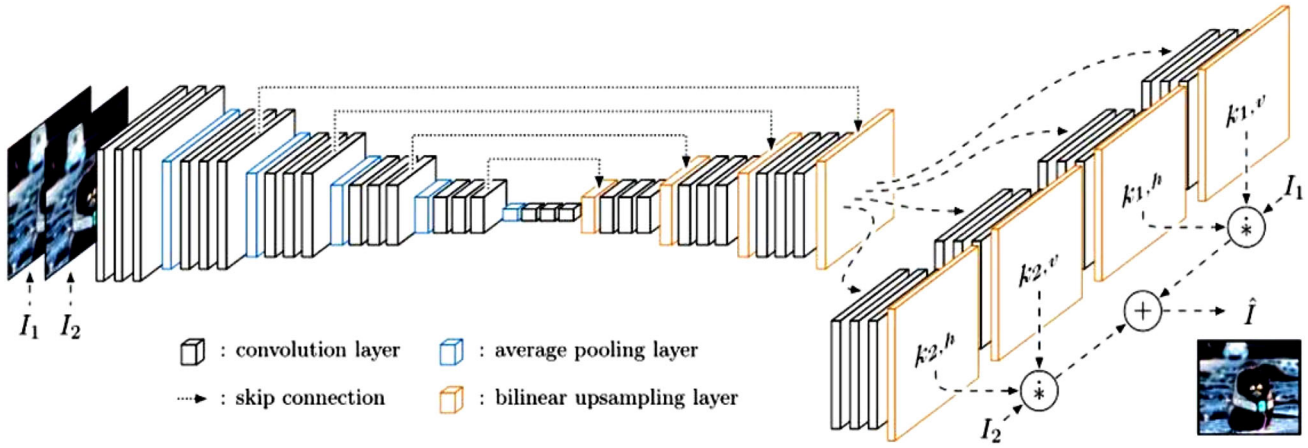


Fig. 4 An overview of adaptive separable convolution neural network architecture [13]. Input frames I_1 and I_2 are processed using encoder–decoder framework to generate four 1D kernels corresponding to every output pixel. The estimated kernels locally convolved with the input frames generate interpolated frame $\hat{I}$. Majority of kernel-based algorithms are inspired by this framework to execute and optimize state-of-the-art video frame interpolation [24]

the kernel size, which is pre-defined in these methods, is less than the actual motion flow of pixels. Secondly, these methods are highly memory extensive. To solve these problems, Cheng et al. (2020) proposed to use more relevant pixels to estimate kernels adaptively calling it deformable separable convolution (DSepConv) [37], hence using smaller kernel size with relevant features for handling large motion. DSepConv uses the encoder–decoder network for feature extraction. These features are used to estimate separable kernels, masks and offsets for each pixel in the frame. Trained on Vimeo-90k [7] DSepConv [37] produce more visually appealing results and is less computationally expensive when compared with other kernel-based methods [5,13]. However, as with other kernel-based methods, it can only generate a single interpolated frame between two consecutive input frames. The authors of DSepConv [37] improved their existing model in EDSC [38]. They were able to reduce the number of parameters to be trained while maintaining the same results. They were also able to generate multiple interpolated frames between two consecutive frames making it the first kernel-based approach to do so. However, the results for arbitrary time interpolation were not as good as state-of-the-art flow-based approaches.

3.2 Flow-based methods

The goal is to determine the nature of flow between corresponding entities in consecutive frames and explicitly synthesize intermediate images to enhance the resulting video quality. High-quality video frame interpolation often depends on precise motion estimation techniques that train mathematical or deep learning models to establish a strong correlation between consecutive frames in order to preserve the continuity of flow, based on the actual optical displacement of flow vectors and trajectory of visual components via relevant occlusion reasoning and color consistency methods [39]. So far, flow-based methods have managed to achieve comparable results parallel to the latest GAN [14], CNN, hybrid technologies, and are evolving evidently to outstand heavy computational requirements of deep learning methods on real-time benchmarks. Based on the genre of flow considered as a baseline for motion interpolation, the flow-based approach is divided into three major subcategories as discussed in Sect. 3.2.1, 3.2.2 and 3.2.3.

3.2.1 Path selective interpolation

The first approach is built upon the intuitive idea that every pixel in the interpolated frames traces out a path in the predecessor frames. The anticipated path justifies the movement of pixel gradients, as described in Fig. 5. With the parallel implication of correspondence and coherence criteria, the most optimal path is obtained by minimizing the energy function and hence chosen as the desired path for that pixel. A prominent feature is its transitioning property over the blending approach preserving original frequency content of images and greatly simplifying occlusions and blur. Unlike standard optical flow and stereo techniques, path computation is more robust in capturing forward and backward flows in un-occluded regions and needless to consider visibility explicitly. Moreover, it eases the identification of occluded regions deterministically as a post-processing operation by only matching flow consistency. Finally, this approach proves to show significant improvement in terms of visual quality over the past decade with advancing works in image gradient operations. However, a broad scope of improvement lies in

exploring more genres of feature points to overcome complex, intense lightning changes. Few popular state-of-the-art interpolation strategies are discussed below.

Dhruv Mahajan et al. (2009) [40] proposed a path framework parallel to an inverse optical flow approach that computes background motion of arbitrary intermediate pixel p in the input frames as shown in Fig. 5. The idea is to move and copy pixel gradients along the anticipated path traced by every pixel from source to the destination image, thus avoiding usual chromatic aberrations produced such as holes or visual blur using standard optical flow methods [41–44]. The transition over the blending approach preserves the frequency details of input frames without ghosting. Also, interpolating gradients instead of actual intensities guards edge preservation of images.

Bo Yan et al. (2013) [45] improved the scheme mentioned above by introducing two leading innovations: first, using standard optical flow [19,36,46] to supervise path direction by constraining path length as illustrated in Fig. 6 and maintaining global path coherency using the Lucas Kanade algorithm [47,48] and second, by employing a pixel interlacing model to significantly optimize the optical flow estimation process for more accurate path selection. In contrast to the original framework [40], narrowing down the solution space of path set significantly improves the efficiency of path construction step and overall efficiency of the algorithm.

Yizhou Fan et al. (2016) [49] further guided the optimization process of path construction by collaborating conventional path-based framework of Mahajan [40] and Bo Yan [45] with useful feature points extracted from input frames [50]. Integrating semantic information identifies critical pixels in input frames. It supervises the method for accurate motion pattern recognition via optimal energy minimization [51], thus avoiding wrong path selection and achieving more natural results as described in Fig. 6. The processing time is limited by constraining maximum cost value $C_{max} = 10^4$ to prevent memory overflow and possible timeouts. Further, the method achieves reasonable performance for bigger size input images and is eligible to produce an arbitrary number of intermediate frames while considering motion propensity supporting high visual quality.

3.2.2 Optical flow guided

The second and most traditional approach to address frame rate upscaling is by utilizing bidirectional optical flow that perceives motion information across consecutive images and captures dense pixel correspondences. The estimated flow guides the warping process to convert input images to the interpolated frame location and constructively blend, maintaining space–time coherency in anticipated motion. Optical flow suggests the apparent motion of objects moving in bi-dimensional motion space, which can be explored as a subproblem of image interpolation domain. Classical optical flow-based approaches adopted a variational model utilizing an energy minimization process. Fast deep neural network-based approaches proposed recently like CNN-based approaches as discussed above in Sect. 3.1 still suffer from two major disadvantages: 1) Separate computation of forward and backward optical flows during bidirectional optical flow estimation neglects the correlation between symmetric optical flows and hinders continuity of sequenced flow and 2) the majority of the latest optical flow networks utilize conventional coarse-to-fine warping frameworks [52,53], which are unable to capture detailed motion and focus widely on large-scale motion only [54,55].

Another area of interest lies in refining the pixel synthesis stage, i.e., blending pixels of warped frames to produce an interpolated frame. Occlusion is handled simultaneously using bidirectional information [56,57], but is constrained to pixel-wise blending. Recent deep learning-based pixel synthesis approaches like Super SloMo [4] and CtxSyn [10] substantially improve performance by persuasively utilizing local information of surrounding pixels, as shown in Fig. 7. However, constant research is conducted to minimize occlusion and hole problem effectively by fusing extra information about the location of holes and color consistency [39,57]. The performance of the methods discussed below relies majorly on the quality of estimated optical flow. As explained in the previous section, the robustness of generating the unknown flow vectors provides a major field of the ongoing investigation in this domain.

Manuel Werlberger et al. (2011) [35] utilized the representation of image sequences in a space–time volume by employing minimization of optical flow driven TV-L1 energy functional, which relies on spatial and flow-guided temporal gradients. Linear movement of pixels is assumed while propagating the optical flow vector that was countered by Dhruv Mahajan [40] by constructing a path-based framework to handle non-rigid complex motion. Inspired by the ROF denoising model, the method supports a wide variety of industrial applications such as reconstructing completely lost frames, restoring impaired frames, image sequence denoising, and frame interpolation.

Lars Lau Raket et al. (2012) [58] proposed to re-parametrize optical flow energy described by Manuel [35] with a symmetric data fidelity term that utilizes both neighboring frames as references. Re-parametrizing original energy functional minimizes the extra work of temporal warping and makes the process of calculating bidirectional flow superfluous. Notably, motion vectors have only half the length of the ones obtained from the regular parametrization making this method better suited to handle large displacements compared to traditional methods that only make use of a one-sided linearization. Convenient implementation on

Table 1 Objective results comparison using state-of-the-art techniques on UCF101, Middlebury, Vimeo-90K, and Xiph

	Category	Vimeo-90k [7]			UCF101 [18]			Xiph-2K			Xiph-"4K"			Middlebury [19]	Parameters
		PSNR ↑	SSIM ↑	LPIPS ↓	PSNR ↑	SSIM ↑	LPIPS ↓	PSNR ↑	SSIM ↑	LPIPS ↓	PSNR ↑	SSIM ↑	LPIPS ↓	IE	(million)
SpyNet [53]	Flow-based	31.95	0.940	0.032	33.67	0.963	0.032	–	–	–	–	–	–	2.49	–
ToFlow [7]	Flow-based	33.73	0.952	0.027	34.58	0.947	0.027	33.93	0.922	0.061	30.74	0.856	0.132	2.51	1.1
EpicFlow [74]	Flow-based	32.02	0.946	0.029	33.71	0.941	0.029	–	–	–	–	–	–	2.47	1.1
PoSNet [84]	Flow-based	34.32	0.953	0.028	34.21	0.948	0.027	–	–	–	–	–	–	–	40.1
Super SloMo [4]	Flow-based	34.75	0.968	0.028	33.14	0.938	0.028	34.60	0.926	0.064	31.82	0.868	0.158	2.28	19.8
MIND [2]	CNN-based	33.50	0.943	0.028	33.93	0.966	0.028	–	–	–	–	–	–	3.35	–
SepConv -$\mathcal{L}_1$ [13]	Kernel-based	33.80	0.956	0.027	34.79	0.947	0.029	34.77	0.929	0.067	32.06	0.880	0.169	2.27	21.6
SepConv -$\mathcal{L}_F$ [13]	Kernel-based	33.45	0.951	0.019	34.69	0.945	0.024	34.47	0.921	0.041	31.68	0.863	0.097	2.44	21.6
PhaseBased [73]	Phase-based	–	–	–	32.45	0.953	0.034	–	–	–	–	–	–	–	–
DVF [1]	CNN-based	31.54	0.921	0.059	34.12	0.946	0.027	–	–	–	–	–	–	4.04	1.6
CyclicGen [6]	CNN-based	32.10	0.923	0.058	35.11	0.950	0.030	33.00	0.901	0.083	30.26	0.836	0.142	2.86	3.0
MEMC-Net [24]	Hybrid	34.29	0.956	0.027	34.96	0.948	0.030	–	–	–	–	–	–	2.10	70.3
DAIN [8]	Hybrid	34.70	0.964	0.022	35.00	0.950	0.028	35.95	0.940	0.084	33.49	0.895	0.170	2.04	24.0
MetaTrained(DAIN) [123]	Hybrid	34.94	.968	0.021	35.04	0.952	0.027	35.98	0.940	0.084	33.52	0.897	0.170	-	24.0
Ctx-Syn -$\mathcal{L}_{Lap}$ [10]	Flow-based	34.39	0.961	0.024	34.62	0.949	0.031	35.71	0.936	0.073	32.98	0.890	0.175	–	–
Ctx-Syn -$\mathcal{L}_f$ [10]	Flow-based	33.76	0.955	0.017	34.01	0.941	0.024	35.16	0.921	0.035	32.36	0.857	0.081	–	–
Softmax -$\mathcal{L}_1$ [88]	Flow-based	**36.10**	0.970	0.021	**35.39**	0.952	0.033	**36.62**	**0.944**	0.107	**33.60**	**0.901**	0.234	-	-
Softmax -$\mathcal{L}_f$ [88]	Flow-based	35.48	0.964	**0.013**	35.10	0.948	**0.022**	35.74	0.921	**0.029**	32.50	0.856	**0.071**	-	-
DSepConv [37]	Kernel-based	34.73	0.974	0.028	35.08	**0.969**	0.030	–	–	–	–	–	–	2.06	21.8
AdaCof [33]	Kernel-based	34.27	0.971	0.031	34.91	0.968	0.029	–	–	–	–	–	–	2.31	21.8
CAIN [124]	CNN-based	34.65	0.973	0.031	34.91	**0.969**	0.032	–	–	–	–	–	–	2.28	42.8
EDSC [38]	Kernel-based	34.84	**0.975**	0.026	35.13	0.968	0.029	–	–	–	–	–	–	**2.02**	8.9

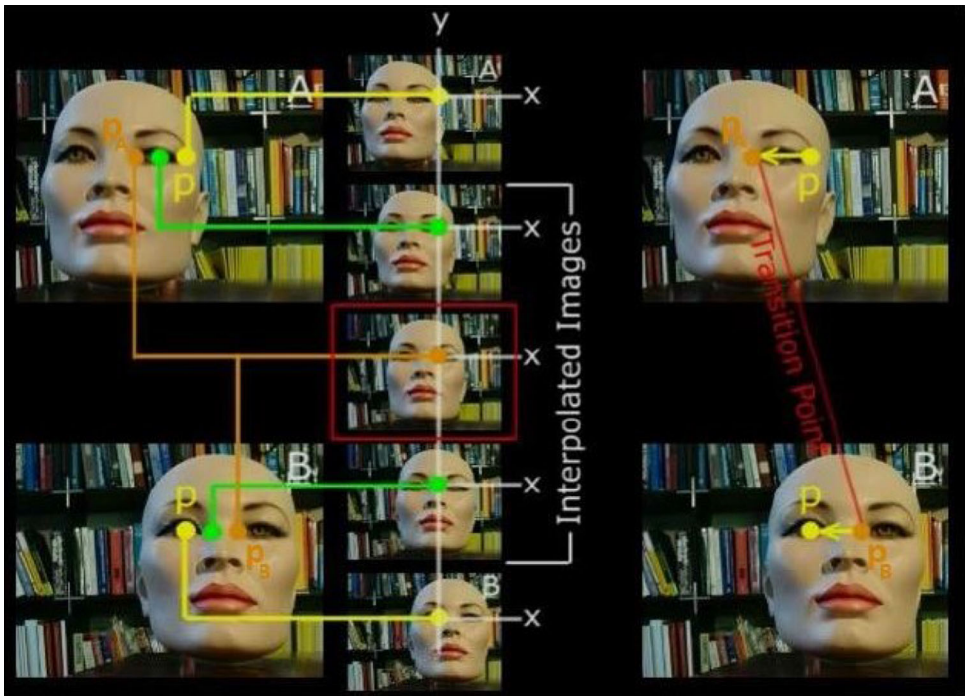


Fig. 5 An illustration of a generic path-based framework. It explains how the path is defined in moving gradients method [40]. Point p in the starting input image travels to transition point p_A and p_B in intermediate images A and B, respectively, and finally at point p in the ending input image. The direction of movement of the pixel is approximately in the opposite direction of the movement of the object

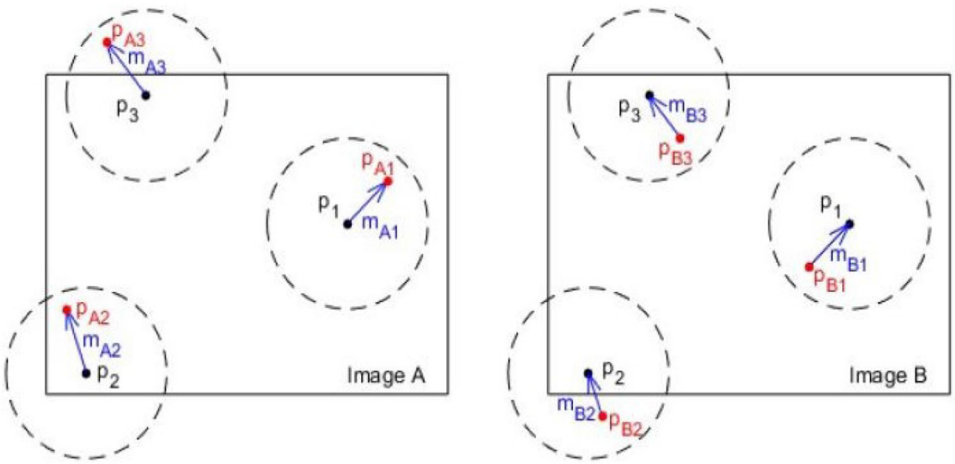


Fig. 6 Path vector selection process. The above figure describes examples of valid and invalid paths. In image A, path vectors m_{A1} and m_{A2} stand valid, while path vector m_{A3} stands invalid. In image B, path vectors m_{B1} and m_{B3} are valid, while path vector m_{B2} is invalid. That proves only optimization trial of pixel p_1 is valid, while trials of p_2 and p_3 are invalid

NVIDIA Tesla C2050 GPU enables real-time frame rate doubling of standard 30fps video footage of resolution 640 × 480.

Hoda Rezaee Kaviani et al. (2015) [59] focused on eliminating severe artifacts like holes, cracks, salt-and-pepper noise [60] in the reconstructed frame by mapping a patch of pixels to a new position in the target frame instead of mapping individual pixels. It outperforms existing state-of-the-art motion-compensated methods [61–64] with an average PSNR increment of about 1-2 dB. Parallel processing of patch-based modules accelerates the execution speed with visually compelling results for faster motion. The technique involves five major ingredients: optical flow motion estimation [61], patch-based reconstruction scheme, mismatch mask generation, decision making step concluded by a hole filling module using IPHI [65]. Efficiency relies majorly on frame-wise pixel count and default parameter values of optical flow motion estimation (ME).

Counter to the above, Wenbin Li et al. (2016) [66] suggested an effective strategy to enhance local smoothness of complex non-rigid motion by adding a Laplacian cotangent mesh constraint [67]. It applies a mesh system with a specificity of one vertex per pixel on every image intending to preserve local geometric details by minimizing angular differences through multiple nested fixed point iterations. Fine-tuning of parameters like vertex density index of an input mesh and smoothness term weight is used that concludes its strong performance against other non-rigid optical

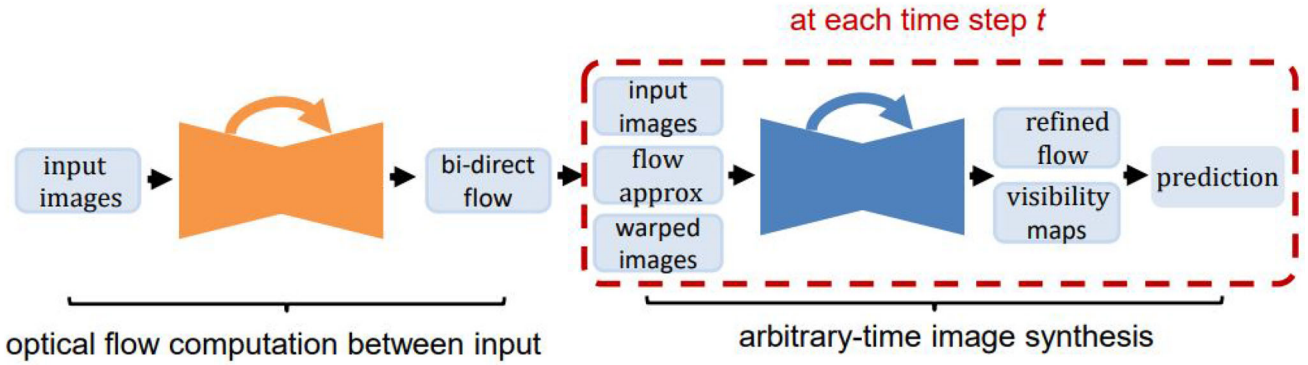


Fig. 7 A standard framework for the majority of optical flow-based interpolation methods is inspired by SuperSloMo [4]. Fully convolutional U-Net architecture estimates bidirectional optical flow and uses it to generate interpolated frame via backward warping and bilinear interpolation. The most traditional approach to address frame rate upscaling is by utilizing bidirectional optical flow that perceives motion information across consecutive images and captures dense pixel correspondences. The estimated flow guides the warping process to convert input images to the interpolated frame location and constructively blend, maintaining space–time coherency in anticipated motion

flow algorithms such as Garg et al.'s [68] spatiotemporal of guided method that utilizes interdependence with neighboring pixels motion to constraint the flow computation field. Quantitatively it excels *Middlebury interpolation error* criteria described in Table 1, and smarter mesh designs can further provide better interpolation approximations for future interests.

Huaizu Jiang et al. (2018) [4] explored the success of deep learning techniques in high-level computer vision tasks that served as an inspiration to solve complex frequency bound motion, as shown in Fig. 7. His work SuperSloMo is an extended version of U-Net architecture proposed by Ziwei Liu et al. [1] to preserve spatial and temporal coherency jointly integrated with occlusion handling framework in a self-supervised manner [69]. The time-independent nature of learned network parameters enables it to produce multiple intermediate frames simultaneously. Training is done with 1132 video clips from real-time cameras and YouTube of 240-fps. The primary purpose of handling complex occlusions is served judiciously as every interpolated pixel mirrors optical flow from either bidirectional flows at a linearly adjacent position in consecutive images. FlowNet2 [70] provides a strong baseline to compute bidirectional optical flows between consecutive frames by effectively avoiding motion boundary blur. Overall, Super SloMo's consistent performance against both non-neural and CNN-based approaches supports its state-of-the-art standards on Middlebury [19], KITTI 2012 benchmark [21], slowflow, high-frame-rate Sintel datasets [71], and UCF101 [18].

Ting Zhang et al. (2018) [72] deviated the heavy dependence on optical flow estimation accuracy by crafting a multi-scale dense network for frame interpolation (FIMSDN) with escalated feature propagation. The constructed network fully exploits multi-scale information for large displacement frame interpolation and outstands its competing methods [13,70,73–75] by distinctively using optical flow after warping but not hinging to it. Precisely, bidirectional flow generated between consecutive input frames using a pretrained FlowNet2 model [70] is utilized to estimate enclosed motion via mapping functions applied to corresponding pixels in both frames. Warping is done spatially midway using computed optical flow, and the interpolated frame is produced directly by supplying original frames into FIMSDN. In contrast to the computationally expensive CNN-based supervised methods, this method is independent of the ground truth of optical flow for training and even not entirely dependent on it for flow estimation module execution.

So far, experts have faced two significant challenges in this domain: first, to accurately capture large-scale fast motion and second, to simplify occlusion reasoning to enhance visual quality. CBOF-Net (2019) [23] supports a two-tier network architecture comprising optical flow evaluation and pixel synthesis sub-modules. Iterative estimation of optical flow [56] caters to preserve the continuity of the optical flow sequence by keeping track of supplementary information into the network and improves its accuracy. Besides, radical development in conventional coarse-to-fine architecture enables us to estimate the motion of fine structures impressively by mitigating the warping effect of forward flow to avoid hole problems. Furthermore, the pixel synthesis network combines statistical information, including color consistency of optical flows and sputtering frequency, to adhere to the occlusion problem effectively [57].

Tejas Javashankar et al. (2019) [76] devised an efficient optical flow estimation method based on the local all-pass algorithm [77] by exploiting high approximation order, typically quadratic. In contrast, conventional optical flow methods use only first order. It is a leading opti-

cal flow-guided state of the art for real-time operation at high spatiotemporal resolutions under critical computational requirements of optical flow estimation network. Overall, subjective and objective results justify the pleasing perceptual quality and smoother interpolated videos favorably comparable to the supreme CNN method [13] and prove LAP as a consistently fair candidate among preferred state of the art.

Catering to address temporal constraints of video camera sensors that assume uniform motion between successive frames and cannot intensively capture fast complex motion, Xiangyu Xu et al. (2019) [78] proposed an acceleration-aware quadratic video interpolation method to render high-quality interpolation results by allowing predictions with curvilinear motion trajectories and variable velocities. In contrast to state-of-the-art approaches based on linear models [1,4,12,13] as illustrated in Fig. 7, the quadratic model provides higher-order video interpolation built on an encoder–decoder network of U-Net [79,80] which effectively estimates and refines flow maps of backward flow fields to suffice accuracy of interpolation results. It is designed in parallel with the multiple nearest neighbor-based interpolation (MNBI) [32] approach that simulates barrel distortion using nearest accurate pixels.

The fundamental aspect of VFI framework as seen till now is to produce smooth motion with minimum visual blur and preserving local information of mobile objects in the produced intermediate frames. This generally involves two strategies, including frame rate upscaling and frame deblurring. New techniques are coming up with a joint video enhancement solution, namely generating a higher rate of frames which are blur-free from initial low-frame-rate hazy input frames. Wang Shen et al. (2019) [81] introduced a blurry VFI technique to process motion blur via EDVR [82] and SNR [83] along with parallel upscaling of frame rate using SuperSloMo [4], MEMC-Net [24] and DAIN [8]. The author incorporates the functionality within a pyramid module that cyclically produces blur-free intermediate frames. The pyramid model improves restoration ability and computational complexity by incorporating flexible temporal scope and spatial receptive field. To enhance this further, an inter-pyramid recurrent module is integrated to exploit temporal dependencies by associating sequential models. This recurrent component enables iterative extraction of temporally smooth intermediate frames with least effect on the size of the model. Its exceptional performance on Adobe240 [20] and Youtube240 datasets can be observed exclusively from Table 2.

The increasing upsurge of optical flow inspired Songhyun et al. (2019) [84] to introduce a VFI framework that converts frame rate to 4× the standard rate using a combination of a flow estimation module coupled with an enhancement network [7,85]. This method is more robust and produces unique

Table 2 Objective results comparison using state of the art on Adobe240 and Youtube240 datasets

	Adobe240 [20]		YouTube240	
	PSNR	SSIM	PSNR	SSIM
Super SloMo [4]	27.52	0.859	30.84	0.910
MEMC-Net [24]	30.83	0.912	34.91	0.959
DAIN [8]	31.03	0.917	35.06	0.96
BIN [81]	32.51	0.928	35.10	0.946
FI-MSAGAN [119]	33.37	0.937	–	–

flow estimators for every possible direction and position of the frame in the network [10]. Earlier methods [4,58,72,86] were based on a single flow estimator as described in Fig. 7 to double the frame rate or generate multiple intermediate frames. In comparison, this method underpins any missing information by utilizing flow maps and given input frames in the flow estimator network as supplementary input to the enhancement network. The intuition behind using two different models to generate three intermediate frames: PosNetS and PoSNetM alleviate artifacts caused by inconsistent optical flow, thus improving overall visual quality especially in regions having occluded pixels and mobile boundaries of constituent objects. The ablation studies confirm that this method successfully performs at real-time execution speed and subjective quality.

The techniques explored so far [78,87] have skillfully used backward warping to render functions like flow estimation and depth prediction as a version of differentiable image sampling. Relatively, forward warping had the least influence due to its inherent challenges such as solving the ambiguity of multiple pixels mapping toward the same position in the output frame. Simon Niklaus et al. (2020) introduced Softmax Splatting [88] to counter this phenomenon shift and apply it effectively for frame interpolation tasks. Two input frames fed as input are forward warped along with their feature pyramid structures using the optical flow of the softmax splatter module. This is continued by a synthesis network that generates an intermittent frame from these warped inputs. To exploit the generalization of U-Nets, it skillfully employs GridNet [11] architecture and mitigates checkerboard rarities by utilizing the advancements laid down by Niklaus et al. [10]. Hence, it effectively predicts intermediate frames at any arbitrary time apart from fine-tuning the optical flow and feature pyramid to show its competence against prominent state of the art as shown in Table 1.

The primary interest of the latest studies inclines toward processing super-high-resolution fast and real-time videos of 4K, 8K, 16K frames in a single pass with minimal resource resetting. So far, state of the art is achieved for standard resolution videos of UCF101 [18] and Middlebury [19] benchmarks using conventional flow-, CNN-, GAN- and

phase-based methods, namely Super SloMo [4], SepConv-$\mathcal{L}_f$ [13], Ahn et al. [17], and DAIN [8], which are less suitable to handle HD videos of 4K due to very limited low-image-resolution training benchmarks. Ha-Eun Ahn et al. (2019) [28] proposed an advanced 4K VFI method that performs multi-scale optical flow refinement to produce intermediate HD frames in a single pass and outperforms efficiency standards by running 4.39× faster than new 4K methods [14,15,17] along with real-time visual quality. The main model is composed of three sub-modules, namely OFE and two coarse-to-fine OFR networks. Novel to previous approaches, it targets to reconstruct optical flow information rather than directly modifying pixel information in order to minimize the visual blur of traditional video enhancement methods [60]. Performance is evaluated on SJTU Media, Ultra Video, and upgraded Vimeo datasets containing suitable 4K image resolution and is not covered in this paper.

3.2.3 Motion-compensated interpolation

The third and most widely used approach to derive intermediate flow is utilizing motion vectors to determine the transformation scheme of a given reference frame to the target frame. These methods were designed to overcome the discrepancies of earlier non-motion-compensated methods of frame repetition and frame averaging that produced impractical results with motion jerkiness and ghosting artifacts to interpolated frames. Hence, advanced FRUC approaches perform frame interpolation along motion trajectories called motion-compensated FRUC methods and can generate higher-quality reconstructed frames in return for more computation. Due to the fast development of technology, higher computational complexity can be tolerated for FRUC.

It is a sequential process containing three major steps called motion estimation (ME), motion vector smoothing, and motion-compensated interpolation (MCI). Motion estimation is programmed to compute the "velocity" vector of each pixel in the input frame, i.e., the trajectory followed by pixel in a temporal unit [47]. Further, estimated motion vectors compensate each pixel spatially halfway to determine constituent motion [89]. However, they are degraded due to misplaced blocks or "tears," resulting in poor qualitative results described commercially as a "soap opera effect." MC techniques are commonly operated using pixel-based or block matching algorithms, the latter being more useful to handle faster motion with less blocky artifacts consistently. Motion estimation for block matching bifurcates to two kinds: unilateral ME where block matching occurs for every block in the previous frame, followed by linkage to block in the next frame, thus interpolating intermediate block at the corresponding location, as discussed by a FRUC method proposed by Jeon et al. [90]. Being more prone to leave holes or overlaps in the intermittent frame, bilateral ME [65] is suitably preferred that adopts temporal symmetry between blocks of consecutive frames.

Earlier approaches discussed in this paper perform numerical computations to solve interpolation problems. These methods suffer from limitations as follows. Firstly, most of them do not consider the spatial consistency of neighboring pixels. Second, to the best of our knowledge, none of them considers and analyzes the reliability of estimated motion trajectories. Furthermore, none of them considers the possibility of employing multiple motion trajectory hypotheses to obtain a better estimation for the intermediate frame. This paper aims to discuss the relative performance of advanced MC techniques with state-of-the-art frame rate up-conversion (FRUC) algorithms.

Hongbin Liu et al. (2012) [92] integrated both temporal motion model and spatial image model to a multi-hypothesis Bayesian FRUC model to rebuild the optimization criterion to predict the interpolated frame of the maximum posterior probability. The model employs a set of "optimal" motion fields to build a group of motion trajectory hypotheses instead of a unique optimal solution, as shown in Fig. 9. The resulting pixels in the interpolated frame are a weighted combination of the reliability of each solution. The method is evaluated to be quite suitable for sequences having a variety of motion scales. Significant yields of PSNR values compensate for surplus performance time by analyzing empirical results, which supports that the outcome of the proposed scheme can achieve a reasonable objective and subjective quality if real-time constraints are not highly critical. An extension of such an algorithm was proposed by Doosep Choi et al. (2015) [93] tending to minimize computation cost vis-à-vis comparable level of performance. The task is accomplished by addressing the issue of multiple protrusive local minima by a maximum posterior probability (MAP)-based MV refinement method that iteratively upgrades true MV estimates of every block by computing a weighted combination of cumulatively estimated locally neighboring MVs and current observed MV according to the unreliability factor stored as locally static additive Gaussian noise (AGN) variance.

Zhefei Yu et al. (2013) [34] studied the distinct behaviors, mutual dependence, and interaction among various levels of adjacent video frames and suggested a self-corrective multi-level model including three of them as pixel level, block level, and sequence level. Constructive algorithms are implemented for each level, i.e., block-level ME by eliminating unreliable MVs, and so on. Productive exploitation by preserving level-wise advantages and learning from selective information to overcome inherent limitations is the core purpose of this algorithm.

Won Hee Lee et al. (2013) [64] proposed an occlusion reasoning-based solution that anticipates four interpolated frames using the reliability of estimated motion vector fields produced by an advanced optical flow framework [56]. A

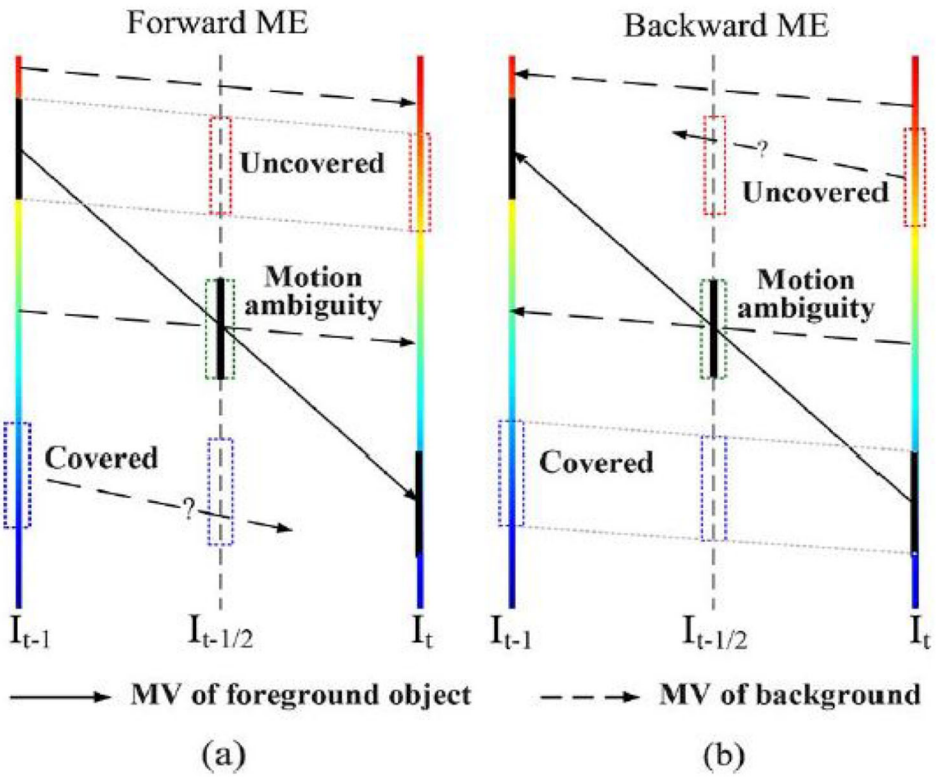


Fig. 8 Illustration of occlusion and motion ambiguity problems as discussed in MCFI [86]. Black and colored region corresponds to foreground objects and background details, respectively. Both the uncovered portion of backward ME and covered portions of forward ME are lost in the target frame which generates false MVs and generates ambiguities in bidirectional motion evaluation. Finally, portions where the correct MVs of background and foreground objects come across each other, and it is tough to select motion for a typical foreground object. Background MVs if chosen produce more visual artifacts due to irregular trajectories of moving objects

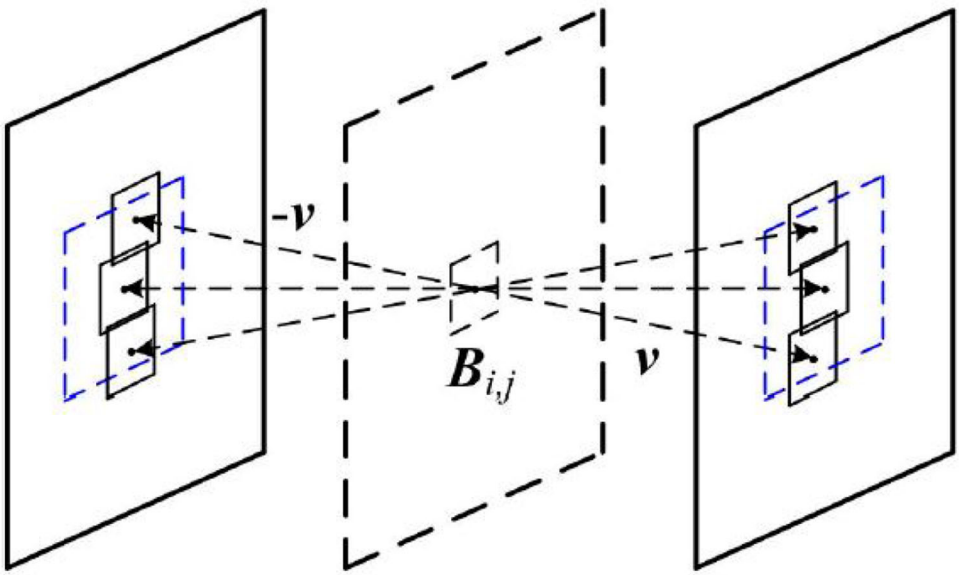


Fig. 9 Full search motion estimation. Search window is a candidate set to find optimal motion trajectory. Computational complexity is huge for full search ME due to large size of candidate set. Advanced successive elimination algorithms are utilized to relax the overall computational cost [91]

combination of the above-said frames produces a singular interpolated entity by utilizing a variational image fusion module by utilizing data energy terms based on differential relationship curve between pixel reliability and error distributions of interpolated images. The method opens excellent possibilities for future work in occlusion handling.

Optical flow methods discussed in the previous section suffer from motion approximation in a limited space or time. As an attempt to overcome shortcomings of block-based approaches, many region-based interpolation approaches have been opted despite implementation complexity [94, 95]. Region-based motion-compensated frame interpolation methods segment the input images into a variable-shaped group of pixels based on pixel intensity and motion homogeneity using image segmentation modules. The computation time is largely occupied for image segmentation, and SAD (Sum of absolute differences) calculation as the region merging process is comparatively less tedious. Hyungjun

Lim et al. (2014) [96] claims the superiority of this method than previous optical flow-based motion-compensated methods by subjectively comparing MV fields of various ME methods and ground truth [46,47,97–100]. As a result, this method provides significantly improved exhaustive search and region-based motion vectors outperform former state-of-the-art methods.

Un Seob Kim et al. (2014) [65] focused on reasonably optimizing real-time performance rendering better PSNR values by adopting a prediction-based motion vector smoothing (PMVS) to efficiently eliminate outliers using MVs of neighboring local blocks, partial average-based motion compensation (PAMC) that simplifies blocking artifacts in intermittent frames by using region-wise partial average and intra-predicted hole interpolation (IPHI) of magnitude H.264/AVC to reduce motion blurriness. Implementation using shift operations on predetermined weights widely relaxes computational complexity. In contrast to earlier bilateral motion estimation algorithms [101,102], the method establishes strong interpolated frame PSNR improved by 3.44 dB on an average and hardly a loss of 0.13 dB than competing unilateral ME employed algorithms. Overall, an effective yield of 89.3% relaxation in computational complexity based on absolute difference proves the importance of the described method.

Qingchun Lu et al. (2016) [86] upgraded the solution to occlusion handling and motion discrepancies. Following the basic principle of temporal motion consistency among multiple consecutive frames, the uncovered, overlapped type of motion ambiguous regions is identified as shown in Fig. 8 to accurately compensate bidirectional MVFs using auxiliary information. Also, an improved version of traditional overlapped block motion estimation (OMBC) is maintained through statistical adjustments to subdue unwanted motion blurring and ghostly artifacts. Compared to previous methods, it claims to render an efficient real-time performance tested in three benchmark methods: Dual ME [62], novel TME [103], and iterative ME [104]. The average PSNR of interpolated frames also shows a proximate increment of up to 1.788 dB.

Yongbing Zhang et al. (2016) [87] formulated the pixel intensity variation across consecutive frames through continuous and differentiable Taylor series functions to minimize discolorations across neighboring frames and maintain motion continuity. The idea conduces to find the most optimal motion vector by employing a motion-aligned partial derivative (MAPD) computation algorithm that guides Taylor approximation to closely match forward and backward polynomial approximations at interpolated frame position. Experimental analysis proves the superiority of described polynomial approximation method over preexisting monomial approximation ME methods [62,92].

Jiang et al. (2017) [105] considered the development of graphic technologies and modeled a prototype for efficiently interpolating 3D videos, generally having a limited frame rate due to huge memory demand. The approach re-computes MVF of the intermittent frame using the BME module, followed by the classification of image blocks into occluded and normal blocks. Post-processing of occlusion blocks through foreground–background segmentation and normal blocks using color–depth information is carried out for interpolation [106]. Simulation studies prove the significant raise in PSNR and SSIM metrics' values than conventional MCFI methods, as shown in Fig. 8 with more natural and fluid interpolated video quality.

Simon Niklaus et al. (2018) applied a context-aware synthesis [10] approach by incorporating per-pixel contextual information to eliminate motion estimation discrepancies and occlusion and synthesize a better, visually appealing interpolated frame. Input images fed to a pre-trained neural network generate context-specific information, which enhances the estimated interpolated frame quality by applying these context maps on output frames of usual optical flow estimation and warping steps. Distinguished from the regular approaches that mix the pre-warped frames, this methodology uses both input frames, and their context maps to a video frame synthesis deep learning model to generate interpolated frame in a context-aware way. Performing motion compensation before interpolation allows adding more frames at temporally arbitrary position $t \in [0,1]$. Unlike other CNN-based approaches that are directed to interpolate at fixed time t, it does not require to retrain its model on changing time t of interpolated frame saving extra computation cost of alternative approaches like recursive interpolation or retraining the pre-defined model.

Li et al. (2019) [91] figured a spatially predictive model for MC-FRUC (SP-MCFI) that splits every incoming frame into two categories of blocks called basic and absent blocks. The optimized version of bilateral motion estimation (BME) through the successive elimination algorithm (SEA) is applied to compute MVs of basic blocks as described in Fig. 9, while MVs of absent blocks are discovered quite accurately using surrounding MVs of neighboring blocks. Reducing the search space of motion vectors using adequate parameter settings greatly contributes to increase computational efficiency.

Zhao et al. (2019) [107] presented an edge-based refinement of estimated MVFs by utilizing edge information in variable block ME module and hole filling in MCI module. Computation overhead of edge-based component is counterbalanced by tangibly good quality visual results compared to conventional optical flow-guided and MSEA method.

Li et al. (2020) [108] designed a low-complex version with advanced EPF that subsamples high-frequency components of video frames to minimize accuracy degradation

before BME. The real-time EPF enforces edge preservation of constituent objects by aiding BME to reduce mismatched blocks by nullifying the poor effects of homogenous structures in texture regions of video frames. BME is further optimized by opting out redundant and irrelevant candidates from search space of conventional FS (Full Search) and predict MV out of spatial and temporal neighbors as illustrated in Fig. 9 conforming to the local smoothness of Motion Vector Field (MVF). Experimental evaluations suggest fruitful subjective and objective gains compared to most recent methods [109,110] at a lower computation cost.

3.3 GAN-based interpolation

With advancing trends of the digital age, the demand for the photorealism of motion media tends to prioritize accurate but smooth results. The state of the art discussed in previous sections notably improved optical flow estimation using deep convolution neural networks and MCI remarkably exploit motion information between consecutive frames, but inherit their artifacts resulting in qualitatively poor performance correlated with hardware constraints. Inspired by the evident success of deep convolutional neural networks in video processing tasks and thorough research in the field of GANs by Goodfellow since 2014, the first GAN interpolation network designed by Mark Koren et al. (FINNiGAN [111]) in 2016 managed to cut traditional "ghosting" and "tearing" artifacts and compensated for fast-motion discrepancies. The general architecture of a GAN framework is illustrated in Fig. 10. Few developments have been discussed below.

The first attempt by Mark Koren et al. (2016) [111] enhanced frame rate by utilizing the convolutional neural network framework collaborated with generative adversarial networks known as FINNiGAN. The idea is to prevent common structural information loss during up-sampling by using a SIN (structure interpolation network) that produces the structure of the intermediate frame constrained by a weighted combination of four significant losses: $\mathcal{L}_1$ loss, clipping loss, *MS-SSIM* loss, and discriminative loss. Color and texture inconsistencies in SIN output were further addressed by pipelining a refinement network that appends GAN loss with $\mathcal{L}_1$ loss to refine output frame quality. Tensorflow supports the efficient implementation of the model. It manages to outperform naïve LFI (*Linear FI*) and DFI (*Deep FI*) results by producing sharper and more structured frames with eliminated checkerboard-style artifacts.

Zhe Hu et al. (2018) [112] proposed a multi-scale structure to share parameters across various layers and cut costly global optimization, unlike usual flow-based methods. MSFSN (multi-scale frame synthesis network) became the first model to provide flexibility by parametrizing the temporal locus of the interest frame. It adopts a more compact network than popular auto-encoder methods [5,13,74] while maintaining comparable reconstruction accuracy. Qualitatively, a pre-trained VGG network bypasses the outlandish reconstructed frames obtained on minimizing pixel-wise MSE loss functions. The coarse-to-fine structure greatly reduces memory overhead for model storage without compromising accuracy and renders useful services in the latest IoT and smartphone applications. Independent of fixed interpolation settings, it can synthesize multiple frames at any intermediate temporal location. Evaluations suggest that it is a suitable choice over CNN-, kernel-, and voxel-based methods, namely FlowNet2.0 [70], EpicFlow [74], DVF [1], and Sep-Conv $\mathcal{L}_f$ [13] where computation resources are limited.

Subject to real-time interpolation, efforts to reduce optimization parameters are always a prime concern of modern developers, thereby cutting excessive hardware costs. Chenguang Li et al. (2018) [113] exploited multi-scale CNN architecture to support the long-varied motion and employed additional WGAN-GP (Wasserstein generative adversarial network loss with gradient penalty) [114] to achieve more natural results. The model has a slim generator network structure requiring relatively less storage and high-speed processing due to residual structure and cumulative generation of high-resolution frames. Instead of image pyramids [112], feature pyramids [115] are built to achieve better visual experience. The loss function is a weighted triplet of $\mathcal{L}_1$, WGAN-GP, and perceptual loss that overcomes the defects of each loss function adopted individually, as discussed earlier. Wei Xue et al. (2019) [116] further utilized this framework to enhance the performance of frame rate upscaling of GAIT videos.

Joost van Amersfoort et al. (2019) [14] remarkably established the most popular state of the art called FIGAN well known for its exemplary performance on real-time YouTube 8M videos with an average runtime speedup of $\times 47$ than immediate competing method [13] and high PSNR gains. The proposed framework is a multi-scale network supervised at various levels with mixed perceptual loss function stating the earliest model to combine the pyramidal system of traditional optical flow modeling with the evolution of spatial transformer networks. The author claims to surpass flow-based and phase-based methods and produce sharper, visually compelling results comparable to SepConv-$\mathcal{L}_f$ at a lower proportion of training parameters under real-time scenarios.

Motivated by the great success of GAN architecture in low-level and high-level computer vision problems, Shiping Wen et al. (2019) [117] laid out a strong network of two concatenated GANs (built on U-Net [79]), former learning motion from training video clips and latter integrating finer frame details to enhance output quality. Counter to standard adversarial losses, it employs Normalized product correlation loss (NPCL [118]) that support its exceptional performance over relatively noisy results of earlier approaches.

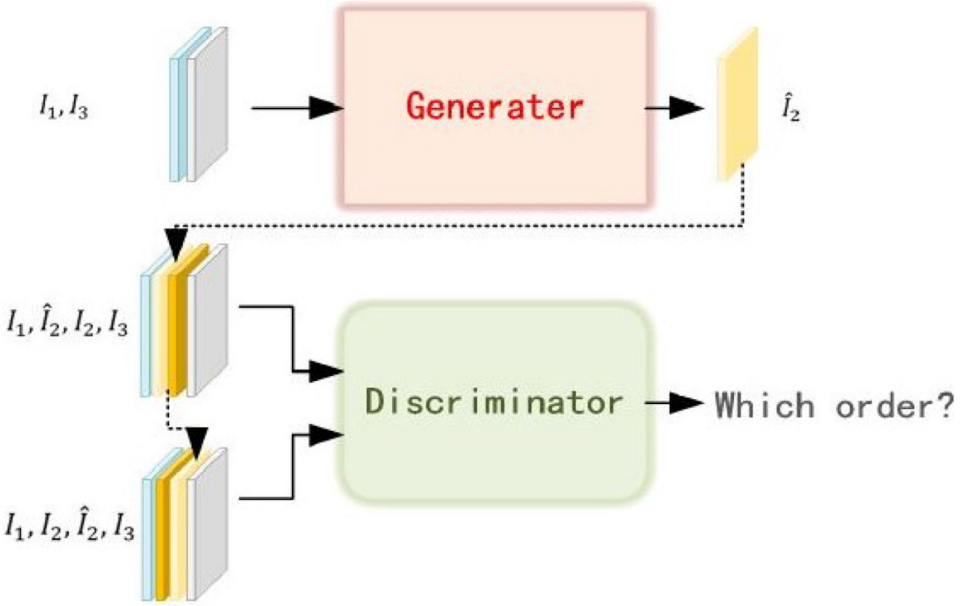


Fig. 10 Basic module of a Generative Adversarial network. The discriminator functions as a classifier to characterize between various input orders produced by the generator

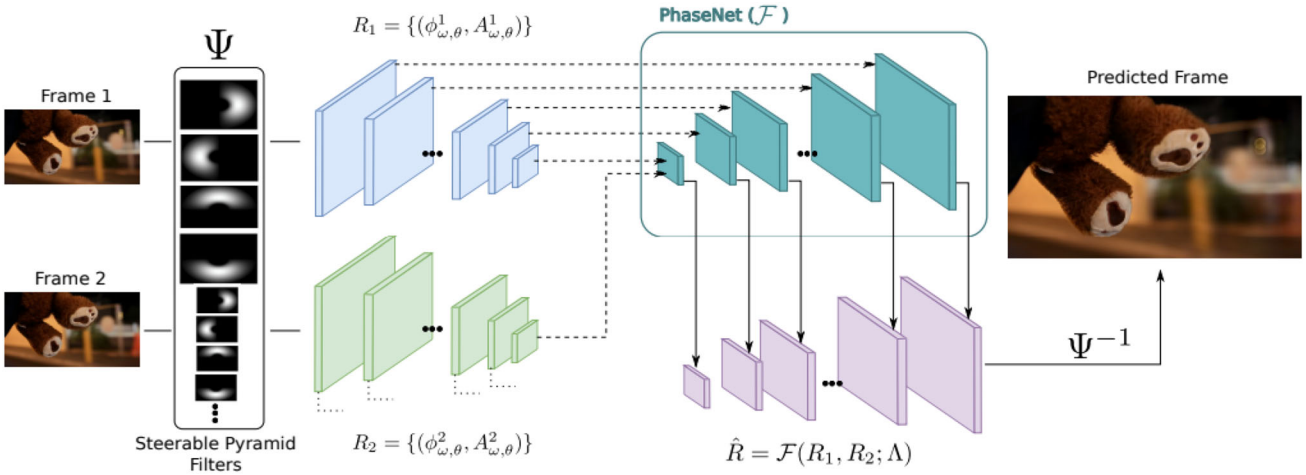


Fig. 11 PhaseNet for video frame interpolation [12] estimates phase shift and amplitude values that were hand-tuned in phase-based interpolation [73]. Their network employed a decoder only architecture to imitate level-wise decomposition of phase information. The decomposition of two given input images shown as R_1 and R_2 is achieved by enacting the steerable pyramid filters (ψ). These are fed as inputs to the PhaseNet, purely based on a decoder framework. The dimensions and count of layers is parallel to the decompositions R_1 and R_2. The features of constituent blocks of each level are available in PhaseNet paper. For a given level of input frame decomposition, the links from the former are illustrated to mitigate frame cluttering. The estimated response from the filter, $\hat{R}$ is utilized to generate the intermediate frame

Xiao et al. [119] interpolated intermediate frames using generative adversarial networks in which they introduced an attention network for focusing on moving objects. The introduced frame interpolation framework using multi-scale dense attention generative adversarial networks, i.e., FI-MSAGAN uses multiple generators and discriminator networks with input images of different sizes for a better combination of local and global information details. Runtime and accuracy for FI-MSAGAN are comparable to other state-of-the-art methods.

3.4 Phase-based methods

The initial work in using phase information for frame interpolation is done by Didyk et al. (2013) [120]. Their method was based on the supposition that phase shift values of each pixel

encode small motion information. However, their method was unable to perform well for large motion. Their method was improved significantly by Meyer et al. (2015) [73]. They used a coarse-to-fine structure to adjust phase shift information. They proposed a multi-scale pyramid level structure to propagate phase information. They put an upper bound on phase shift to accommodate large motions. Their algorithm consists of calculating phase shift, interpolate using phase difference, and blending interpolated frame using amplitude values. The phase shift value is calculated at each level of the pyramid with an upper bound. They assume that both large and small motion occurs at comparable frequencies. Their method fails at high frequencies. Even areas with small motion but at high frequencies appear to be blurred. The above approaches contain hand-tuned parameters for image generation. Both phase shift value and amplitude value were calculated. Meyer et al. (2018) [12] proposed a network to estimate phase shift and amplitude values. Their network can thus handle a larger range of motion and frequencies. Their network used a decoder only architecture as described in Fig. 11 to imitate level-wise decomposition of phase information. All layers are identical except for the last layer. The resolution of the interpolated frame increases level by level. The parameters that were hand-tuned in phase-based [73] approach were directly estimated using Phase-Net.

3.5 Other hybrid methods

Several methods did not fall into any of the categories or combine one or more approaches. Ahn et al. (2019) [28] employed a multi-scale motion reconstruction network in their video frame interpolations method. This technique first estimates bidirectional optical flow in a lower resolution than the input frame. For 4K videos, they use one-fourth of the resolution for estimated bidirectional optical flow. Then, they recreate the estimated optical flow for the original resolution by employing a multi-scale reconstruction scheme that can recreate high optical resolution stably. They used multi-scale smoothening loss, consistency loss, adversarial loss, and more to train their network. The proposed network can be divided into three sub-networks. First is an optical flow estimation (OFE) network that estimates low-resolution bidirectional optical flow in a computationally efficient manner. The other two are multi-scale optical flow reconstruction (OFR) networks. They reconstruct optical flow in the original resolution from the low-resolution optical flow estimated by the OFE network. It shows computationally better results than those methods that are operable for 4K videos with comparable visual quality.

Kim et al. (2019) [121] realized that modern video requirements include not only high frame rates but also high resolutions. While [28] takes advantage of high-resolution frames, it does not increase frame resolution. [121] proposed a joint model that not only interpolates frames but also increases the spatiotemporal resolution of frames.

There are two major approaches for deep learning-based video frame interpolation, that is, motion compensation and motion estimation. These methods either estimate convolutional kernels for motion compensation as shown in Fig. 4 or estimate flow and then warp the input frames for motion estimation. Various approaches have been discussed in the earlier sections. Kernel-based methods [5,13] are computationally expensive, and flow-based methods often produce blurry results [60]. Bao et al. (2018) [24] attempted to combine the two approaches. Both convolution kernels and flow vectors are generally estimated using CNN. They proposed a method to combine the two approaches by introducing an adaptive warping layer which used flow vectors and motion compensation kernels to generate output pixels. This layer can also be employed for other video enhancement techniques like super-resolution.

For handling large complex motion, several other techniques employ a coarse-to-fine strategy like Deep Voxel Flow [1] or adopt advanced flow estimation architecture like in context-aware paper [10] using PWC-net, or calculate an occlusion mask for adaptively blending the pixels like in Super SloMo [4], MEMC-Net [24], or interpolation kernels to adaptively generate output pixels from a large neighborhood like in SepConv [13]. While other methods rely on the network to handle occlusion, usually by training with a large amount of data, Bao et al. (2019) [8] handled occlusion explicitly by using depth information. MEMC-Net implicitly handles the occlusion by estimating occlusion masks, extracting contextual features. In contrast, DAIN explicitly detects occlusion.

DAIN [8] relies on a straightforward observation that nearer objects ought to be synthesized first within the intermediate frame (calculate the contribution of every flow vector supported the depth price for aggregation). DAIN consists of the subsequent sub-modules: the flow estimation, context extraction, depth estimation, kernel estimation, and frame synthesis networks as visible from Fig. 12. In DAIN, a depth-aware flow projection layer is employed to synthesize intermediate flows that ideally sample nearer objects than farther ones. It also learns hierarchical features from neighboring pixels to gather contextual information. Then it uses an adaptive warping layer to exploit the optical flow effectively, local interpolation kernels, contextual features, and depth maps to synthesize interpolated frames. For estimating depth maps from the input frames, DAIN employed the model of Chen et al. [122] which is an hourglass network trained on MegaDepth dataset. Compared to the MEMC-Net, DAIN uses 69% fewer parameters. More accurate depth maps can be obtained by predicting depth maps from input frames and modeling the consistency between depth maps and optical flow [9]. Their model is efficient and compact

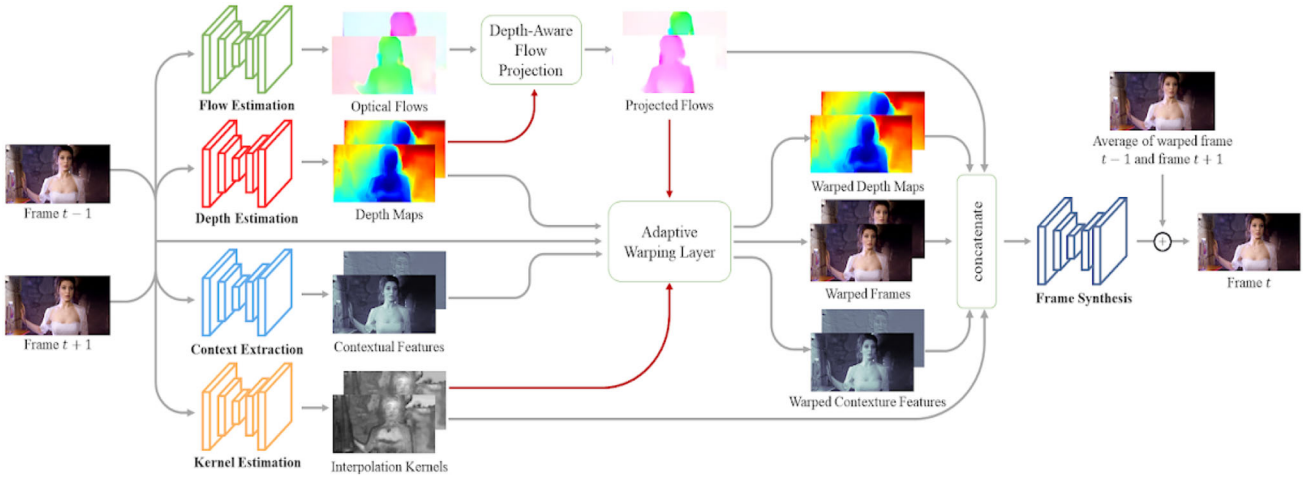


Fig. 12 Depth-aware video frame interpolation architecture [8]. In DAIN, a depth-aware flow projection layer is employed to synthesize intermediate flows that ideally sample nearer objects than farther ones. It also learns hierarchical features from neighboring pixels to gather contextual information. Then it uses an adaptive warping layer to exploit the optical flow effectively, local interpolation kernels, contextual features, and depth maps to synthesize interpolated frames

and performs reasonably well against other existing frame interpolation methods as evident from the performance metric values in Table 1.

Choi et al. [123] showed the benefits of test time adaptation of the network in video frame interpolation tasks through meta-learning strategy. Their scene-adaptive frame interpolation technique adapts to unseen new videos at test time to achieve a significant amount of improvements in the interpolated frames. This strategy can update the weights/parameters of any existing frame interpolation models using just frames present at the test time. This is the first implementation of the meta-learning technique in video interpolation domain. By incorporating meta-learning technique at test time, performances of base models like DVF [1], SuperSloMo [4], SepConv [13], and DAIN [8] have improved. Hence, without any change in the architecture of existing video frame interpolation methods, the scene-adaptive frame interpolation algorithm can be employed.

Choi et al. [124] and Xiao et al. [119] utilized neural networks which focus on important regions of the feature representations, i.e., attention networks for interpolating video frames effectively. Instead of explicit optical flow estimation, Choi et al. [124] reply on channel attention. They utilize channel attention method proposed in Zhang et al. 2018b [125] for video frame interpolation framework. Trained on Vimeo-90k [7], CAIN (channel attention for frame interpolation) utilizes feature reshaping operation (PixelShuffle) with channel attention as a replacement for optical flow computation module. CAIN, when compared with other state-of-the-art methods, is efficient in terms of both time and memory consumption.

4 Discussion

4.1 Performance analysis

We compare different video interpolation techniques on several benchmark datasets including UCF101 [18], MiddleBurry [19], Vimeo-90k [7], and Xiph[2]. We interpolate the intermediate frame at t = 0.5 temporal location in comparative experiments here. For quantitative analysis, we used four comparison metrics. An important note about these metrics is that they are calculated based on the generated images in comparison with the ground truth. First one is peak signal-to-noise ratio (PSNR) (see Eq. 1). It is a common metric used for measuring image quality. PSNR is defined as the ratio of maximum power of signal to the power of noise signal. For color images, i.e., images having three RGB values per pixel, PSNR can be defined via MSE which is sum over all squared value differences divided by image size and number of channels. For an $m \times n$ image with 3 channels where I is ground truth and K is interpolated frame, its PSNR value is given by:

$$PSNR = 20.log_{10}(MAX_I) - 10.log_{10}(MSE) \tag{1}$$

where MAX_IMAX I is maximum possible pixel value and MSE is mean squared error given by:

$$MSE = \frac{1}{3mn} \sum_{i=0}^{m-1} \sum_{j=0}^{n-1} [I(i,j) - K(i,j)]^2 \tag{2}$$

[2] https://media.xiph.org/video/derf

PSNR is used as a quality measurement between the reconstructed and original image. The higher the ratio, the better the quality of the reconstructed image. Second is Structural Similarity (SSIM) index (see Eq. 3). It is used for measuring image quality by measuring the perceptual difference between two similar images. SSIM index on two windows x and y of common size $N \times N$ is given by:

$$SSIM(x, y) = \frac{(2\mu_x\mu_y + c_1)(2\sigma_{xy} + c_2)}{(\mu_x^2 + \mu_y^2 + c_1)(\sigma_x^2 + \sigma_y^2 + c_2)} \quad (3)$$

where, μ denotes average of the window σ^2 denotes variance of the window σ_{xy} denotes covariance of window x and y and c_1 and c_2 are stabilizing constants.

SSIM cannot judge which is better but just how much they differ from each other. SSIM is based on visible structures in the image. The third is the Middlebury interpolation error (IE). One particular advantage of interpolation error is that it does not need ground truth. It is calculated by using the optical flow to extrapolate the current frame. The extrapolated image is then compared with the real next frame of the video.

The last one is learned perceptual image patch similarity (LPIPS) metric. It derives from the observation that deep network activations can be effectively used as a perceptual similarity metric. For our experiments, we used the version 0.1 of LPIPS. In contrast to PSNR and SSIM, the lower value of LPIPS indicates better results.

We test various video frame interpolation techniques to compare the techniques based on the metrics described in this section. We tested the approaches with the same testing dataset for all the approaches whose open source implementation was completely available. For some approaches, we copied the performance results as given by the author under the confirmation that they are tested with the same dataset.Some missing entries in the table can be accounted to the fact that the results were not available for them or we were unable to test that particular method on that dataset. In Table 1 , we provide the quantitative results of various video frame interpolation techniques. Each approach has been appropriately tagged with the category they fall into as described in this paper. Along with PSNR, SSIM, and LPIPS values on Vimeo-90K [7], UCF101 [18], and Xiph, we also provide interpolation error (IE) in the table. The number of parameters to be trained is a governing metric to the size of the model and the time required for training. Therefore, we included the number of parameters in the table. Similarly, in Table 2, we provided the quantitative results on Adobe [20] and YouTube dataset.

In this paper, we presented a comprehensive study of a variety of models for video frame interpolation. We compared them based upon PSNR, SSIM, and LPIPS values on UCF101 [18], Vimeo-90k [7], and Xiph evaluation datasets and interpolation error (IE) on Middlebury dataset [19]. Better comparison values do not always necessarily mean visually pleasing results; a visual comparison is necessary. Although visual inspection cannot be used directly as a metric, as it is subjective, it can reveal artifacts and other distortions which are hard to measure with quantitative methods.

4.2 Challenges

The emergence of low-cost deep learning frameworks that productively exploit the color and motion information of high-quality video sequences has significantly inspired great developments in interpolation processing methods. Promising outcomes of deep learning approaches, on a set of constrained datasets such as UCF101 [18], Vimeo-90k [7] and Middlebury [19]. Despite this success, results are a lot more to be achieved to satisfy real-time requirements. In fact, it is quite tedious to design an efficient, intelligent interpolation system. Such an idea poses numerous challenges.

Encoding spatial and temporal information As discussed, several strategies are available to skillfully capture intermediate motion between frames to synchronize temporal and spatial information of the interpolated frame effectively. We employ deep fully convolutional neural network to estimate pixel-wise spatially adaptive kernels [5,13], or predict spatially and temporally regularized between optical flow that preserves local correlations using convolutions [1,30,70] over temporal irregularities of non-deep learning approaches [74], or utilize pixel-wise phase shift [12,73] to determine motion information, or accommodating corresponding spatial weights in above feature networks [1,4,7,14,24,32]. However, all these methods have their associated drawbacks. Temporal fusion process tends to bypass the temporal sequence; 3D pooling filters and 3D filters have a very stringent temporal structure, so they tend to intake a fixed number of frames as input that is always insufficient; optical flow computation is generally costly and involves side effects such as visual glitches due to edge distortion, depth inconsistencies, abrupt brightness changes, and others. Modeling spatiotemporal coherence of frames stands a major challenge in the interpolation tasks.

Scenario-specific training Most of the advanced deep learning strategies demand in-depth labeled training data. However, in real-time scenarios, gathering and cleaning high-definition video data are laborious and memory-intensive, especially in the medical research field. For example, 4K VFI techniques [17,28] are built on efficient platforms to avoid system collapse during training. It is observed that fine-tuning parameters of interpolation networks with spatial information is more beneficial than training all over again. Several data augmentation tacts are utilized to lead robust scenarios, and overfitting is controlled by calibrating

the learning rate. Another aspect involves the limited quality of results due to domain-specific training datasets [7,18,19]. Learned models tend to perform better over similar video graphic scenarios, which limits its robustness to produce an optimum quality of high-frame-rate videos for a wider variety of generalized data. However, effectively training deep learning networks from a contingent form of training data remains a challenging future assignment.

Visual artifacts and occlusion These between frames estimated using depth maps [8,9] might cause viewpoint variation that generates ambiguous results for repetitive arrivals of the same actions, and occlusion may lead to loss of graphic details. Many commonly used datasets constrain subjects to perform actions in a restricted and visible background to get rid of occlusion, and eventually, this leads to less occluded but limited view data collection. However, interactions in practical scenarios are bound to have occlusion, which makes it challenging to segregate entities in overlapping regions, as illustrated in Fig. 8 and extract features of individual objects, resulting in the ineffectiveness of various existing approaches. A possible solution to figure out occlusion and viewpoint variation involves operating on multi-sensor systems [39]. Such systems can procure multi-view data, with a downside of synchronization requirement and recognition/feature fusion within different views. This adds up to the computation cost and processing complexity. Numerous methods have been proposed to deal with occlusion and viewpoint variation. Evan Herbst et al. [57] opted for a bidirectional flow-based optical flow algorithm parallel to spatial regularizations to handle occlusions and dis-occlusions. However, expertly training deep learning networks to handle occlusion remains is a constant challenge.

Intra-action localization We come across numerous incidents that require exact spatiotemporal localization of suspicious/semantically significant events via interpolation modeling of constituent objects' trajectories. Predicting meaningful constituent action by increasing frame rate is a classic computer vision problem applied widely in fields of human behavior recognition and activity analysis, motion-based sports, art rehearsal, video retrieval, and many others. There are two fundamental challenges to this task: first, identification of subtle inherent characteristics of movements of entities in various scenarios that may justify intervening motion; and second, carrying out predictions as fast as possible in the demanding social world, with a limited set of prior observations. This becomes more complicated provided real-time challenges, e.g., background clutter, occlusion, fast large-scale motion. Accurately predicting the specific intermediate events has a wide range of applications while taking prior decisions in health care, surveillance, and autonomous robots. How to develop convincing algorithms in this direction is a potential concern.

5 Conclusion

This paper puts forward a comprehensive survey of classic video frame interpolation techniques using deep learning. We present a broad overview of existing widely used benchmark evaluations. The available techniques are divided into five major categories given their modality: flow-based, CNN-based, phase-based, GAN-based, and hybrid methods. The five modalities exhibit their unique features and branch to diverse choices of deep learning techniques to utilize their properties productively. The inherent spatial, temporal, and structural attributes of a video sequence are identified. From the aspect of spatiotemporal structural encoding, we highlight the pros and cons of available techniques. Based on key insights underlined by the survey, the problem of video frame interpolation contains promising research opportunities. Performing frame interpolation in real time would be the main focus of future research work. Incorporating some image enhancement technique with real-time frame interpolation can lead to a vast number of practical and useful applications. Furthermore, new techniques in deep learning will enlarge the scope of improvement of frame interpolation. GAN-based training paradigms show a promising future in the field of frame interpolation. Combining frame interpolation with other video processing tasks also seems to interest researchers. It can be said that a wide plethora of research opportunities remain in this domain despite the advances covered till date.

Funding In this study, there is no funding involved from any agency.

Compliance with ethical standards

Conflict of interest The authors declare that they have no conflicts of interest.

Human and animal rights This article does not contain any studies involving humans or animals performed by any of the authors.

References

1. Liu, Z., Yeh, R.A., Tang, X., Liu, Y., Agarwala, A.: Video frame synthesis using deep voxel flow. In: 2017 IEEE International Conference on Computer Vision (ICCV) pp. 4473–4481 (2017)
2. Long,G., Kneip, L., Alvarez, J.M., Li, H., Zhang, X., Yu, Q.: Learning image matching by simply watching video. In: ECCV (2016)
3. Madhawa, V., Sudasingha, I., Vidanapathirana, J., Kanchana, P., Perera, I.: Tracking and frame-rate enhancement for real-time 2D human pose estimation. Vis. Comput **36**, 1501–1519 (2019)
4. Jiang, H., Sun, D., Jampani, V., Yang, M.H., Learned-Miller, E., Kautz, J.: Super slomo: high quality estimation of multiple intermediate frames for video interpolation. In: 2018 IEEE/CVF

Conference on Computer Vision and Pattern Recognition pp. 9000–9008 (2018)
5. Niklaus, S.,Mai, L., Liu, F.: Video frame interpolation via adaptive convolution. In: 2017 IEEE Conference on Computer Vision and Pattern Recognition (CVPR) pp. 2270–2279 (2017)
6. Liu, Y., Liao, Y.T., Lin, Y.Y., Chuang, Y.Y.: Deep video frame interpolation using cyclic frame generation. In: AAAI (2019)
7. Xue, T., Chen, B., Wu, J., Wei, D., Freeman, W.: Video enhancement with task-oriented flow. Int. J. Comput. Vis. **127**, 1106–1125 (2018)
8. Bao,W., Lai, W.S., Ma, C., Zhang, X., Gao, Z., Yang, M.H.: Depthaware video frame interpolation. In: 2019 IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR) pp. 3698–3707 (2019)
9. Yanke, W., Zhong, F., Peng, Q., Qin, X.: Depth map enhancement based on color and depth consistency. Vis. Comput **30**, 1157–1168 (2013)
10. Niklaus, S., Liu, F.: Context-aware synthesis for video frame interpolation. In: 2018 IEEE/CVF Conference on Computer Vision and Pattern Recognition pp. 1701–1710 (2018)
11. Fourure, D., Emonet, R., Fromont, É,Muselet, D., Trémeau, A., Wolf, C.: Residual conv-deconv grid network for semantic segmentation. arXiv:1707.07958 (2017)
12. Meyer, S., Djelouah, A., McWilliams, B., Sorkine-Hornung, A., Gross, M., Schroers, C.: Phasenet for video frame interpolation. In: 2018 IEEE/CVF Conference on Computer Vision and Pattern Recognition pp. 498–507 (2018)
13. Niklaus, S., Mai, L., Liu, F.: Video frame interpolation via adaptive separable convolution. In: 2017 IEEE International Conference on Computer Vision (ICCV) pp. 261–270 (2017)
14. van Amersfoort, J.R., Shi, W., Acosta, A., Massa, F., Totz, J., Wang, Z., Caballero, J.: Frame interpolation with multi-scale deep loss functions and generative adversarial networks. arXiv:1711.06045 (2017)
15. Peleg, T., Szekely, P., Sabo, D., Sendik, O.: Im-net for high resolution video frame interpolation. In: 2019 IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR) pp. 2393–2402 (2019)
16. Jia, Y., Shelhamer, E., Donahue, J., Karayev, S., Long, J.,Girshick, R.B., Guadarrama, S., Darrell, T.: Caffe: convolutional architecture for fast feature embedding. arXiv:1408.5093 (2014)
17. Ahn, H.E., Jeong, J., Kim, J.W.: A fast 4k video frame interpolation using a hybrid task-based convolutional neural network. Symmetry **11**, 619 (2019)
18. Soomro, K., Zamir, A., Shah,M.: Ucf101: A dataset of 101 human actions classes from videos in the wild. arXiv:1212.0402 (2012)
19. Baker, S., Scharstein, D., Lewis, J., Roth, S., Black, M.J., Szeliski, R.: A database and evaluation methodology for optical flow. Int. J. Comput. Vis. **92**, 1–31 (2007)
20. Su, S., Delbracio, M., Wang, J., Sapiro, G., Heidrich, W., Wang, O.: Deep video deblurring for hand-held cameras. In: 2017 IEEE Conference on Computer Vision and Pattern Recognition (CVPR) pp. 237–246 (2017)
21. Geiger, A., Lenz, P., Urtasun, R.: Are we ready for autonomous driving? the kitti vision benchmark suite. In: 2012 IEEE Conference on Computer Vision and Pattern Recognition pp. 3354–3361 (2012)
22. Perazzi, F., Pont-Tuset, J., McWilliams, B., Van Gool, L., Gross, M., Sorkine-Hornung, A.: a benchmark dataset and evaluation methodology for video object segmentation. In: 2016 IEEE Conference on Computer Vision and Pattern Recognition (CVPR) pp. 724–732 (2016)
23. Gu, D., Wen, Z., Cui, W., Wang, R., Jiang, F., Liu, S.: Continuous bidirectional optical flow for video frame sequence interpolation. In: 2019 IEEE International Conference on Multimedia and Expo (ICME) pp. 1768–1773 (2019)
24. Bao, W., Lai, W.S., Zhang, X., Gao, Z., Yang, M.H.: Memc-net: motion estimation and motion compensation driven neural network for video interpolation and enhancement. IEEE Trans. Pattern Anal. Mach. Intell. (2019). https://doi.org/10.1109/TPAMI.2019.2941941
25. Cheng, X., Chen, Z.: A multi-scale position feature transform network for video frame interpolation. IEEE Trans. Circuits Syst. Video Technol. (2019). https://doi.org/10.1109/TCSVT.2019.2939143
26. Mathieu, M., Couprie, C., LeCun, Y.: Deep multi-scale video prediction beyond mean square error. CoRR arXiv:1511.05440 (2016)
27. Zhang, H., Wang, R., Zhao, Y.: Multi-frame pyramid refinement network for video frame interpolation. IEEE Access **7**, 130610–130621 (2019)
28. Ahn, H.E., Jeong, J., Kim, J., Chul Kwon, S., Yoo, J.S.: A fast 4k video frame interpolation using a multi-scale optical flow reconstruction network. Symmetry **11**, 1251 (2019)
29. Krizhevsky, A., Sutskever, I., Hinton, G.E.: Imagenet classification with deep convolutional neural networks. In: CACM(2017)
30. Dosovitskiy, A., Fischer, P., Ilg, E., Häusser, P., Hazirbas, C., Golkov, V., Smagt, P.V.D., Cremers, D., Brox, T.: Flownet: Learning optical flow with convolutional networks. In: 2015 IEEE International Conference on Computer Vision (ICCV) pp. 2758–2766 (2015)
31. Xue, T., Wu, J., Bouman, K., Freeman, B.: Visual dynamics: probabilistic future frame synthesis via cross convolutional networks. arXiv:1607.02586 (2016)
32. Choi, C., Lee, H., Yi, J.: An interpolation method for strong barrel lens distortion. Vis. Comput **34**, 1479–1491 (2017)
33. Lee, H., Kim, T., Chung, T. Y., Pak, D., Ban, Y., Lee, S.: AdaCoF: adaptive collaboration of flows for video frame interpolation. In: 2020 IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR) pp. 5315–5324 (2020)
34. Yu, Z., Li, H., Wang, Z., Hu, Z., Chen, C.: Multi-level video frame interpolation: exploiting the interaction among different levels. IEEE Trans. Circuits Syst. Video Technol. **23**, 1235–1248 (2013)
35. Werlberger, M., Pock, T., Unger, M., Bischof, H.: Optical flow guided tv-l1 video interpolation and restoration. In: EMMCVPR (2011)
36. Barron, J.L., Fleet, D.J., Beauchemin, S.: Performance of optical flow techniques. Int. J. Comput. Vis. **12**, 43–77 (1994)
37. Cheng, X., Chen, Z.: Video frame interpolation via deformable separable convolution. In: AAAI (2020)
38. Cheng, X., Chen, Z.: Multiple video frame interpolation via enhanced deformable separable convolution. arXiv:2006.08070 (2020)
39. Zolfaghari, M., Ghanei-Yakhdan, H., Yazdi, M.: Real-time object tracking based on an adaptive transition model and extended kalman filter to handle full occlusion. Vis. Comput **36**, 701–715 (2019)
40. Mahajan, D., Huang, F., Matusik, W., Ramamoorthi, R., Belhumeur, P.: Moving gradients: a path-based method for plausible image interpolation. In: SIGGRAPH 2009 (2009)
41. Álvarez, L., Deriche, R., Papadopoulo, T., Pérez, J.S.: Symmetrical dense optical flowestimation with occlusions detection. In: ECCV (2002)
42. Xiao, J., Cheng, H., Sawhney, H., Rao, C., Isnardi, M.A.: Bilateral filtering-based optical flow estimation with occlusion detection. In: ECCV (2006)
43. Zitnick, C.L., Jojic, N., Kang, S.B.: Consistent segmentation for optical flow estimation. In: Tenth IEEE International Conference on Computer Vision (ICCV'05) Volume 1, Vol. 2, 1308–1315 (2005)

44. Brox, T., Bruhn, A., Papenberg, N., Weickert, J.: High accuracy optical flow estimation based on a theory for warping. In: ECCV (2004)
45. Yan, B., Chen, Y.: Low complexity image interpolation method based on path selection. J. Vis. Commun. Image Represent. **24**, 661–668 (2013)
46. Horn, B., Schunck, B.: Determining optical flow artificial intelligence **17** (1981)
47. Lucas, B.D., Kanade, T.: An iterative image registration technique with an application to stereo vision. In: IJCAI (1981)
48. Bouguet, J.Y.: Pyramidal implementation of the lucas kanade feature tracker. Open Source Computer Vision Library (2003)
49. Fan, Y., Yoda, N., Igarashi, T., Ma, H.: Path-based image sequence interpolation guided by feature points. In: 2016 IEEE International Conference on Image Processing (ICIP) pp. 569–573 (2016)
50. Zhou, M., Liang, L., Sun, J., Wang, Y.: AAM based face tracking with temporal matching and face segmentation. In: 2010 IEEE Computer Society Conference on Computer Vision and Pattern Recognition pp. 701–708 (2010)
51. Boykov, Y., Veksler, O., Zabih, R.: Fast approximate energy minimization via graph cuts. IEEE Trans. Pattern Anal. Mach. Intell. **23**, 1222–1239 (2001)
52. Sun, D., Yang, X., Liu, M.Y., Kautz, J.: Pwc-net: Cnns for optical flow using pyramid, warping, and cost volume. In: 2018 IEEE/CVF Conference on Computer Vision and Pattern Recognition pp. 8934–8943 (2018)
53. Ranjan, A., Black, M.J.: Optical flow estimation using a spatial pyramid network. In: 2017 IEEE Conference on Computer Vision and Pattern Recognition (CVPR) pp. 2720–2729 (2017)
54. Xu, L., Jia, J., Matsushita, Y.: Motion detail preserving optical flow estimation. IEEE Trans. Pattern Anal. Mach. Intell. **34**, 1744 (2012)
55. Steinbrücker, F., Pock, T., Cremers, D.: Large displacement optical flow computation without warping. In: 2009 IEEE 12th International Conference on Computer Vision pp. 1609–1614 (2009)
56. Baker, S., Scharstein, D., Lewis, J., Roth, S., Black, M.J., Szeliski, R.: A database and evaluation methodology for optical flow. Int. J. Comput. Vis. **92**(1), 1–31 (2011)
57. Herbst, E., Seitz, S., Baker, S.: Occlusion reasoning for temporal interpolation using optical flow. Department of Computer Science and Engineering, University of Washington, Tech. Rep. UW-CSE-09-08-01 (2009)
58. Rakêt, L., Roholm, L., Bruhn, A., Weickert, J.: Motion compensated frame interpolation with a symmetric optical flow constraint. In: ISVC (2012)
59. Kaviani, H.R., Shirani, S.: Frame rate upconversion using optical flow and patch-based reconstruction. IEEE Trans. Circuits Syst. Video Technol. **26**, 1581–1594 (2016)
60. Cosmin, A., Haber, T., Mertens, T., Bekaert, P.: Video enhancement using reference photographs. Vis. Comput **24**, 709–717 (2008)
61. Liu, C., et al.: Beyond pixels: exploring new representations and applications for motion analysis. Ph.D. thesis, Massachusetts Institute of Technology (2009)
62. Kang, S.J., Yoo, S., Kim, Y.: Dual motion estimation for frame rate up-conversion. IEEE Trans. Circuits Syst. Video Technol. **20**, 1909–1914 (2010)
63. Keller, S.H., Lauze, F., Nielsen, M.: Temporal super resolution using variational methods. In: High-Quality Visual Experience, pp. 275–296. Springer, Berlin (2010)
64. Lee, W.H., Choi, K., Ra, J.B.: Frame rate up conversion based on variational image fusion. IEEE Trans. Image Process. **23**, 399–412 (2014)
65. Kim, U.S., Sunwoo, M.H.: New frame rate up-conversion algorithms with low computational complexity. IEEE Trans. Circuits Syst. Video Technol. **24**, 384–393 (2014)
66. Li, W., Cosker, D.: Video interpolation using optical flow and Laplacian smoothness. arXiv:1603.08124 (2017)
67. Li, W., Cosker, D., Brown, M., Tang, R.: Optical flow estimation using Laplacian mesh energy. In: 2013 IEEE Conference on Computer Vision and Pattern Recognition pp. 2435–2442 (2013)
68. Garg, R., Roussos, A., Agapito, L.: Robust trajectory-space tv-l1 optical flow for non-rigid sequences. In: EMMCVPR (2011)
69. Patraucean, V., Handa, A., Cipolla, R.: Spatio-temporal video autoencoder with differentiable memory. arXiv:1511.06309 (2015)
70. Ilg, E., Mayer, N., Saikia, T., Keuper, M., Dosovitskiy, A., Brox, T.: Flownet 2.0: evolution of optical flow estimation with deep networks. In: 2017 IEEE Conference on Computer Vision and Pattern Recognition (CVPR) pp. 1647–1655 (2017)
71. Janai, J., Güney, F., Wulff, J., Black, M.J., Geiger, A.: Slow flow: exploiting high-speed cameras for accurate and diverse optical flow reference data. In: 2017 IEEE Conference on Computer Vision and Pattern Recognition (CVPR) pp. 1406–1416 (2017)
72. Zhang, T., Bai, H., Li, F., Zhao, Y.: Optical flow-guided multiscale dense network for frame interpolation. In: 2018 Asia-Pacific Signal and Information Processing Association Annual Summit and Conference (APSIPA ASC) pp. 1061–1065 (2018)
73. Meyer, S.,Wang, O., Zimmer, H., Grosse,M., Sorkine-Hornung, A.: Phase-based frame interpolation for video. In: 2015 IEEE Conference on Computer Vision and Pattern Recognition (CVPR) pp. 1410–1418 (2015)
74. Revaud, J.,Weinzaepfel, P.,Harchaoui, Z., Schmid, C.: Epicflow: edge-preserving interpolation of correspondences for optical flow. In: 2015 IEEE Conference on Computer Vision and Pattern Recognition (CVPR) pp. 1164–1172 (2015)
75. Farnebäck, G.: Two-framemotion estimation based on polynomial expansion. In: SCIA (2003)
76. Jayashankar, T.,Moulin, P., Blu, T., Gilliam, C.: Lap-based video frame interpolation. In: 2019 IEEE International Conference on Image Processing (ICIP) pp. 4195–4199 (2019)
77. Gilliam, C., Blu, T.: Local all-pass filters for optical flow estimation. In: 2015 IEEE International Conference on Acoustics, Speech and Signal Processing (ICASSP) pp. 1533–1537 (2015)
78. Li, S., Xu, X., Pan, Z., Sun, W.: Quadratic video interpolation for VTSR challenge. In: 2019 IEEE/CVF International Conference on Computer Vision Workshop(ICCVW) pp. 3427–3431 (2019)
79. Ronneberger, O., Fischer, P., Brox, T.: U-net: Convolutional networks for biomedical image segmentation. In: MICCAI (2015)
80. Xu, X., Sun, D., Liu, S., Ren, W., Zhang, Y., Yang, M.H., Sun, J.: Rendering portraitures from monocular camera and beyond. In: ECCV (2018)
81. Shen, W., Bao, W., Zhai, G., Chen, L., Min, X., Gao, Z. .: Blurry video frame interpolation. In: 2020 IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR) pp. 5113–5122 (2020)
82. Wang, X., Chan, K.C.K., Yu, K., Dong, C., Loy, C.C.: EDVR: Video restoration with enhanced deformable convolutional networks. In: 2019 IEEE/CVF Conference on Computer Vision and Pattern Recognition Workshops (CVPRW) pp. 1954–1963 (2019)
83. Xin Tao, Gao, H., Wang, Y., Shen, X., Wang, J., Jia, J.: Scale recurrent network for deep image deblurring. In: 2018 IEEE/CVF Conference on Computer Vision and Pattern Recognition pp. 8174–8182 (2018)
84. Yu S., Park, B., Jeong, J.: Posnet: 4x video frame interpolation using position-specific flow. In: 2019 IEEE/CVF International Conference on Computer VisionWorkshop (ICCVW) pp. 3503–3511 (2019)

85. M. Haris, Shakhnarovich, G., Ukita, N.: Recurrent backprojection network for video super-resolution. In: 2019 IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR) pp. 3892–3901 (2019)
86. Lu, Q., Xu, N., Fang, X.: Motion-compensated frame interpolation with multiframe-based occlusion handling. J. Display Technol. **12**, 45–54 (2016)
87. Zhang, Y., Xu, L., Ji, X., Dai, Q.: A polynomial approximation motion estimation model for motion-compensated frame interpolation. IEEE Trans. Circuits Syst. Video Technol. **26**, 1421–1432 (2016)
88. Niklaus, S., Liu, F.: Softmax splatting for video frame interpolation. In: 2020 IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR) pp. 5436–5445 (2020)
89. Krishnamurthy, R., Woods, J., Moulin, P.: Frame interpolation and bidirectional prediction of video using compactly encoded optical-flow fields and label fields. IEEE Trans. Circuits Syst. Video Technol. **9**, 713–726 (1999)
90. Jeon, B.W., Lee, G., Lee, S., Park, R.: Coarse-to-fine frame interpolation for frame rate up-conversion using pyramid structure. IEEE Trans. Consumer Electron. **49**, 499–508 (2003)
91. Li, Y., Ma, W., Han, Y.: A spatial prediction-based motion-compensated frame rate up-conversion. Future Internet **11**, 26 (2019)
92. Liu, H., Xiong, R., Zhao, D., Ma, S., Gao, W.: Multiple hypotheses bayesian frame rate up-conversion by adaptive fusion of motion-compensated interpolations. IEEE Trans. Circuits Syst. Video Technol. **22**, 1188–1198 (2012)
93. Choi, D., Song, W., Choi, H., Kim, T.: Map-based motion refinement algorithm for block-based motion-compensated frame interpolation. IEEE Trans. Circuits Syst. Video Technol. **26**, 1789–1804 (2016)
94. Jacobson, N., Lee, Y.L., Mahadevan, V., Vasconcelos, N., Nguyen, T.: A novel approach to fruc using discriminant saliency and frame segmentation. IEEE Trans. Image Process. **19**, 2924–2934 (2010)
95. Wang, J., Patel, N., Grosky, W.: Video frame rate up conversion using region based motion compensation. In: 2004 IEEE Electro/Information Technology Conference pp. 143–157 (2004)
96. Lim, H., Park, H.W.: A region-based motion-compensated frame interpolation method using a variance-distortion curve. IEEE Trans. Circuits Syst. Video Technol. **25**, 518–524 (2015)
97. Lim, H., Kim, D.Y., Park, H., Cho, J., Park, S.H., Kim, J.: Motion estimation with adaptive block size for motion-compensated frame interpolation. In: 2012 Picture Coding Symposium pp. 325–328 (2012)
98. Lim, H., Park, H.W.: A symmetric motion estimation method for motion-compensated frame interpolation. IEEE Trans. Image Process. **20**, 3653–3658 (2011)
99. Choi, B.D., Han, J.W., Kim, C.S., Ko, S.: Motion-compensated frame interpolation using bilateral motion estimation and adaptive overlapped block motion compensation. IEEE Trans. Circuits Syst. Video Technol **17**, 407–416 (2007)
100. Zhai, J., Yu, K., Li, J., Li, S.: A low complexity motion compensated frame interpolation method. In: 2005 IEEE International Symposium on Circuits and Systems, Vol. 5 pp. 4927–4930 (2005)
101. Kang, S.J., Cho, K.R., Kim, Y.H.: Motion compensated frame rate up-conversion using extended bilateral motion estimation. IEEE Trans. Consumer Electr **53**, 1759 (2007)
102. Kang, S.J., Yoo, D., Lee, S., Kim, Y.: Multiframe-based bilateral motion estimation with emphasis on stationary caption processing for frame rate up-conversion. IEEE Trans. Consumer Electr. **54**, 1830 (2008)
103. ShashiKiran, S., SrinivasBabu, N., Divakaran, R., MohmadAshiq, A., Arunkumar, B., Rohit, J.: True-motion estimation algorithm and its application to motion-compensated temporal frame interpolation. Int. J. Innovat. Res. Electr. Electron. Instrum. Contr. Eng. **5**, 297–307 (2017)
104. Kim, D.Y., Lim, H., Park, H.: Iterative true motion estimation for motion-compensated frame interpolation. IEEE Trans. Circuits Syst. Video Technol. **23**, 445–454 (2013)
105. Jiang, Y., Yang, X., Feng, Z., Xia, Y.: An efficient 3d video frame interpolation method using color-depth-motion information. In: 2017 4th International Conference on Information, Cybernetics and Computational Social Systems (ICCSS) pp. 77–80 (2017)
106. Yang, X., Liu, J., Sun, J., Lee, Y., Nguyen, T.: Depth-assisted frame rate up-conversion for stereoscopic video. IEEE Signal Process. Lett. **21**, 423–427 (2014)
107. Zhao, Y., Ge, G., Sun, Q.: Frame rate up-conversion based on edge information. In: 2019 7th International Conference on Information, Communication and Networks (ICICN) pp. 158–162 (2019)
108. Li, R., Ma, W., Li, Y., You, L.: A low-complex frame rate upconversion with edge-preserved filtering. Electronics **9**, 156 (2020)
109. Bao, W., Zhang, X., Chen, L., Ding, L., Gao, Z.: High-order model and dynamic filtering for frame rate up-conversion. IEEE Trans. Image Process. **27**, 3813–3826 (2018)
110. Van, X.H.: Statistical search range adaptation solution for effective frame rate up-conversion. IET Image Proc. **12**, 113–120 (2018)
111. Koren, M., Menda, K., Sharma, A.: Frame interpolation using generative adversarial networks. (2017)
112. Hu, Z., Ma, Y., Ma, L.: Multi-scale video frame-synthesis network with transitive consistency loss. 1–12. arXiv:1712.02874 (2017)
113. Li, C., Gu, D., Ma, X., Yang, K., Liu, S., Jiang, F.: Video frame interpolation based on multi-scale convolutional network and adversarial training. In: 2018 IEEE Third International Conference on Data Science in Cyberspace (DSC) pp. 553–560 (2018)
114. Gulrajani, I., Ahmed, F., Arjovsky, M., Dumoulin, V., Courville, A.C.: Improved training of wasserstein gans. In: NIPS (2017)
115. Lin, T.Y., Dollár, P., Girshick, R.B., He, K., Hariharan, B., Belongie, S.J.: Feature pyramid networks for object detection. In: 2017 IEEE Conference on Computer Vision and Pattern Recognition (CVPR) pp. 936–944 (2017)
116. Xue, W., Ai, H., Sun, T., Song, C., Huang, Y., Wang, L.: Framegan: increasing the frame rate of gait videos with generative adversarial networks. Neurocomputing **380**, 95–104 (2020)
117. Wen, S., Liu, W., Yang, Y., Huang, T., Zeng, Z.: Generating realistic videos from keyframes with concatenated gans. IEEE Trans. Circuits Syst. Video Technol. **29**, 2337–2348 (2019)
118. Li, J., Shen, X.: Image blocking parallel processing approaches to a normalized product correlation image matching algorithm. Mini-Micro Syst, **25**(11) (2004)
119. Xiao, J., Bi, X.: Multi-scale attention generative adversarial networks for video frame interpolation. IEEE Access **8**, 94842–94851 (2020)
120. Didyk, P., Sitthi-amorn, P., Freeman, W., Durand, F., Matusik, W.: Joint view expansion and filtering for automultiscopic 3d displays. ACM Trans. Graph. **32**, 221:1–221:8 (2013)
121. Kim, S.Y., Oh, J., Kim, M.: FISR: deep joint frame interpolation and super-resolution with a multi-scale temporal loss. In: AAAI (2020)
122. Chen,W., Fu, Z., Yang, D., Deng, J.: Single-image depth perception in the wild. In: NIPS (2016)
123. Myungsub C., Choi, J., Baik, S., Kim, T., Lee, K.M.: Scene adaptive video frame interpolation via meta-learning. In: 2020 IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR) pp. 9441–9450 (2020)
124. Choi, M., Kim, H., Han, B., Xu, N., Lee, K. M.: Channel attention is all you need for video frame interpolation. In: AAAI Conference on Artificial Intelligence, pp. 10 663–10 671 (2020)

125. Zhang, Y., Li, K., Li, K., Wang, L., Zhong, B., Fu, Y.: Image super-resolution using very deep residual channel attention networks. arXiv:1807.02758 (2018)

Anil Singh Parihar received the B.Tech. and M.E. degrees in electronics and communication engineering and the Ph.D. degree in the area of applications of soft computing in image processing. He is currently working as an Associate Professor with the Department of Computer Science and Engineering, Delhi Technological University, Delhi, India. His research interest includes machine learning, computer vision, pattern recognition, soft computing, and evolutionary algorithms.

Disha Varshney is currently pursuing her B.Tech. degree in computer science and engineering from Delhi Technological University, Delhi, India. Her research interests include applications of deep learning to computer vision problems, pattern recognition, machine learning, and natural language processing.

Kshitija Pandya is currently pursuing his B.Tech. degree in computer science and engineering from Delhi Technological University, Delhi, India. His research interests include applications of deep learning to computer vision problems, pattern recognition, machine learning, and natural language processing. He is also part of Google Developer Student Club DTU core team.

Ashray Aggarwal is currently pursuing his B.Tech. degree in computer science and engineering from Delhi Technological University, Delhi, India. His research interests include computer vision, pattern recognition, natural language processing, and cryptography.