International Journal of Scientific Research in Engineering and Management (IJSREM)
Volume: 10 Issue: 11 | Nov - 2026 SJIF Rating: 8.659 ISSN: 2582-3930

Enhancing Temporal Resolution of Satellite Imagery using AI/ML based Optical Flow

Akanksha Kardile[1], Kaustubh Sonawane[2], Amruta Waghmode[3], Yashwant Pandhare[4], Prof. [Urmila.B.Shelake][5],Hod.[Dr.H.B.Jadhav][6]

[1]Student, Department of Computer Engineering, [Adsul's Technical Campus], [Ahilyanagar], India, akankshakardile9@gmail.com
[2]Student, Department of Computer Engineering, [Adsul's Technical Campus], [Ahilyanagar], India, kaustubhsonawane00@gmail.com
[3]Student, Department of Computer Engineering, [Adsul's Technical Campus], [Ahilyanagar], India, amrutawaghmode53@gmail.com
[4]Student, Department of Computer Engineering, [Adsul's Technical Campus], [Ahilyanagar], India, yashwantpandhare37@gmail.com
[5]Guide, Department of Computer Engineering, [Adsul's Technical Campus], [Ahilyanagar], India, urmilashelake1@gmail.com
[6]Hod, Department of Computer Engineering, [Adsul's Technical Campus], [Ahilyanagar], India, hem3577@gmail.com

---

Abstract - Geostationary weather satellites such as GOES-16 and INSAT-3D usually deliver a full-disk image only every 10 to 30 minutes, which is too coarse for tracking fast convective storms. This paper presents a software-only system that converts 30-minute infrared satellite sequences into 5-minute sequences by generating intermediate frames with deep optical flow. Infrared frames from the SEVIR dataset, NOAA GOES-16/18 archives and INSAT-3D data are cleaned, resized to 256 × 256 pixels and normalised. A Real-Time Intermediate Flow Estimation (RIFE) network, fine-tuned on satellite frame sequences, predicts bidirectional intermediate flows and a fusion mask for any timestep, and the two input frames are backward-warped and fused to produce five new frames between each real pair. Because SEVIR provides real frames every 5 minutes, every generated frame can be scored against ground truth using PSNR, SSIM and MAE at 10, 20 and 30-minute input gaps. The proposed model is compared with linear blending, Farneback optical flow and Super SloMo under the same data split. Results and frame sequences are stored in MongoDB and served through a MERN web dashboard that plays real and generated frames side by side with clear labels.

Key Words: satellite imagery, temporal interpolation, optical flow, video frame interpolation, RIFE, remote sensing.

1. INTRODUCTION

Weather satellites in geostationary orbit observe the same part of the Earth continuously, which makes them the main source of cloud imagery for forecasting and disaster response. Their temporal resolution, however, is limited by how long the imager takes to scan a full disk. The Advanced Baseline Imager on GOES-16 produces a full-disk image every 10 minutes in its default mode, and the Indian INSAT-3D imager provides a full-disk scan roughly every 30 minutes. Convective cells, squall lines and the clouds that feed flash floods can grow, merge or dissipate within this window. A forecaster looking at two frames 30 minutes apart has to guess what happened in between.

Increasing the scan rate needs new hardware or a change of scan strategy, both of which are expensive and outside the control of most users. A software approach is to synthesise the missing frames from the frames that already exist. The simplest version, a weighted average of the two neighbouring frames, gives poor results because clouds move rather than fade. Averaging two shifted cloud fields produces a double image with soft ghost edges. Motion-based interpolation estimates where each pixel travels between the two frames and warps the images along that motion, so clouds are moved rather than blended.

Classical optical flow methods such as Horn-Schunck [5] and Farneback [4] assume brightness constancy and small, smooth motion. Cloud fields break these assumptions: they deform, grow and change brightness as they cool, and upper-level clouds can move tens of pixels in 30 minutes. Deep video frame interpolation models such as Super SloMo [2] and RIFE [1] learn flow from data and handle larger motion, but they are trained on ordinary RGB video of people, cars and scenery. Infrared satellite images are single-channel brightness temperature fields with very different textures and statistics.

This work adapts RIFE to infrared geostationary imagery and builds a complete pipeline around it. The main contributions are: (i) a preprocessing pipeline that reads NetCDF and HDF5 satellite products and prepares consistent training sequences; (ii) fine-tuning of RIFE on satellite frames for arbitrary-timestep interpolation, producing a 6× increase in temporal resolution from 30-minute input; (iii) an evaluation protocol that uses real 5-minute SEVIR frames as ground truth at 10, 20 and 30-minute gaps, with a fair comparison against three baselines; and (iv) a MERN web dashboard that shows real and generated frames side by side and reports per-frame scores.

The rest of the paper is organised as follows. Sec. 2 reviews related work. Sec. 3 describes the proposed system and the interpolation model. Sec. 4 gives the experimental setup, Sec. 5 presents the results and discussion, and Sec. 6 concludes the paper.

2. RELATED WORK

2.1 Classical Optical Flow

Horn and Schunck [5] formulated optical flow as an energy minimisation problem that combines a brightness constancy term with a global smoothness term. Farneback [4] approximates the neighbourhood of each pixel with a quadratic polynomial and estimates dense displacement from the change

in polynomial coefficients between frames. Both methods are fast, need no training data and are available in OpenCV, which makes them common baselines. Their weakness is large or non-rigid motion, where the local linearisation fails and the flow field becomes noisy or over-smoothed.

2.2 Deep Optical Flow and Frame Interpolation

FlowNet [8] showed that a convolutional network can regress optical flow directly from an image pair. PWC-Net [9] added feature pyramids, warping and cost volumes, and RAFT [10] introduced recurrent refinement over all-pairs correlation, which is now a strong reference for flow accuracy. For frame interpolation, kernel-based methods such as adaptive separable convolution [11] predict a local filter for each output pixel, while flow-based methods warp the inputs. Super SloMo [2] computes bidirectional flow between the inputs with a U-Net [17], approximates the intermediate flows linearly and refines them with a second network together with visibility maps. DAIN [12] adds depth to handle occlusion. RIFE [1] estimates the intermediate flows $F_{t\to0}$ and $F_{t\to1}$ directly with a coarse-to-fine network called IFNet, which removes the linear motion approximation and runs in real time on a consumer GPU.

2.3 Interpolation of Satellite Imagery

Vandal and Nemani [13] trained a task-specific optical flow network for temporal interpolation of multispectral GOES-16 imagery and showed that learned flow outperforms linear interpolation across bands. Work on precipitation nowcasting, such as ConvLSTM [16], addresses the related problem of extrapolating future frames rather than filling gaps between known frames. The SEVIR dataset [3] made this kind of study easier by providing aligned radar and satellite sequences sampled every 5 minutes over storm events in the United States. Its 5-minute sampling is what makes direct ground-truth evaluation of interpolated frames possible in the present work. Comparatively little work has evaluated modern real-time interpolation models such as RIFE on infrared satellite data, or applied them to Indian satellite products from MOSDAC.

3. PROPOSED SYSTEM

3.1 System Architecture

Fig. 1 shows the overall architecture. The system has four layers: data ingestion and preprocessing, the interpolation engine, evaluation, and the web application. The interpolation engine and evaluation module are written in Python with PyTorch [15] and OpenCV. The web application follows the MERN stack: MongoDB stores frame metadata and scores, a Node.js and Express server exposes REST endpoints, and a React front end displays the sequences. Inference requests from the dashboard are passed to a Python inference service over HTTP, so the trained model does not need to be ported to JavaScript.

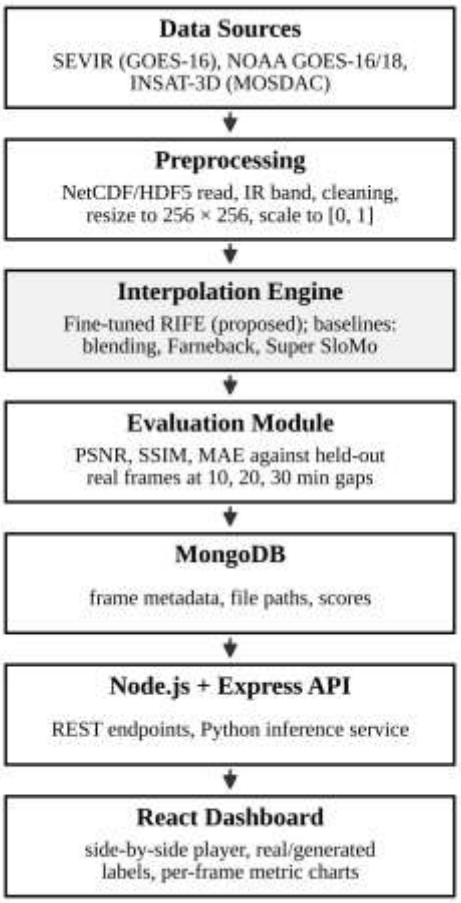


Fig -1: System architecture of the proposed interpolation system

3.2 Data Sources and Preprocessing

Three sources are used, summarised in Table 1. SEVIR [3] is the main training and test source because it provides GOES-16 infrared frames every 5 minutes. Each SEVIR event covers 4 hours as 49 frames over a 384 km × 384 km patch; the infrared channels ir069 (6.9 µm, water vapour) and ir107 (10.7 µm, clean longwave window) are stored at 2 km resolution as 192 × 192 pixel images. NOAA GOES-16/18 Level-1b and Level-2 files in NetCDF format are used for additional sequences outside the SEVIR events. INSAT-3D thermal infrared (TIR1, 10.8 µm) products in HDF5 format are downloaded from MOSDAC [14] to test how the model behaves on Indian data with a 30-minute cadence.

Table -1: Data sources used in this work

Source	Sensor / band	Format	Cadence
SEVIR	GOES-16 ABI, ir069 and ir107	HDF5	5 min
NOAA archive	GOES-16/18 ABI, band 13	NetCDF	10 min (full disk)
MOSDAC	INSAT-3D Imager, TIR1	HDF5	30 min

Preprocessing reads each file with the h5py and netCDF4 libraries, converts the stored counts to brightness temperature in kelvin, and replaces missing or out-of-disk pixels by nearest valid values. Brightness temperatures are clipped to the range 180 K to 330 K and scaled linearly to [0, 1]. Frames are resized to 256 × 256 pixels with bilinear interpolation so that all sources share one input size. Since RIFE expects three-channel input, the single infrared channel is repeated three times, and the mean of the three output channels is taken as the predicted frame. Sequences with missing frames, corrupted scans or strong

calibration jumps are discarded. The data is split by event into training (70%), validation (15%) and test (15%) sets, so that frames from one storm never appear in more than one set.

3.3 RIFE-based Intermediate Frame Generation

Given two input frames I_0 and I_1 and a timestep $t \in (0, 1)$, RIFE estimates the flows from the unknown target frame to both inputs, $F_{t\to0}$ and $F_{t\to1}$, together with a soft fusion mask M with values in [0, 1]. The estimation is done by IFNet, a stack of three IFBlocks working at 1/4, 1/2 and full resolution. Each block receives the inputs, the timestep and the warped results of the previous block and outputs a residual update of the flows and the mask. The process is shown in Fig. 2.

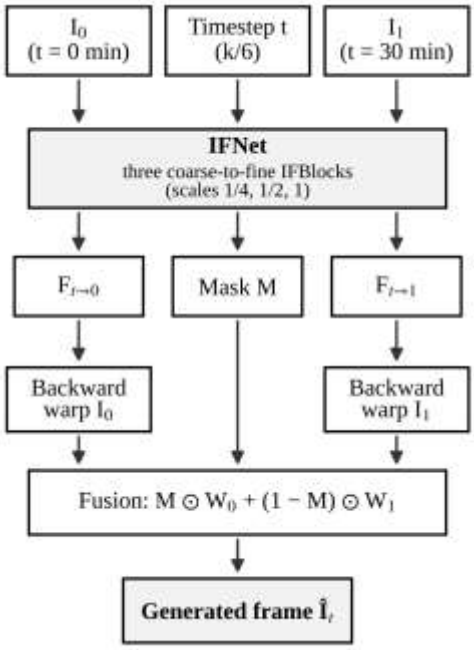


Fig -2: Intermediate frame generation with IFNet

The input frames are backward-warped to time t using bilinear sampling, written here as $W(\cdot)$:

$$\hat{W}_0 = W(I_0, F_{t\to0}), \quad \hat{W}_1 = W(I_1, F_{t\to1}) \tag{1}$$

The generated frame is the mask-weighted fusion of the two warped frames:

$$\hat{I}_t = M \odot \hat{W}_0 + (1 - M) \odot \hat{W}_1 \tag{2}$$

where $\odot$ denotes element-wise multiplication. The mask lets the model take each pixel mainly from the input in which it is visible, which matters at the edges of growing or dissipating clouds. For a 30-minute input pair the model is run five times with $t = k/6$, $k = 1, 2, ..., 5$, producing frames at 5, 10, 15, 20 and 25 minutes. Together with the two real frames this gives a 5-minute sequence, which is a 6× gain in temporal resolution. Fig. 3 shows this layout.

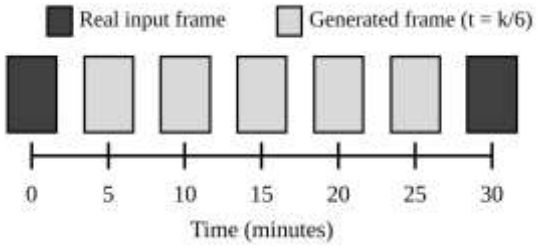


Fig -3: Real and generated frames in one 30-minute interval

3.4 Fine-tuning on Satellite Frames

Training starts from the publicly released RIFE weights trained on Vimeo-90K, so the network already knows general motion patterns. Training samples are built from SEVIR sequences: for an input gap of g frames (g = 2, 4 or 6, that is 10, 20 or 30 minutes), frames i and i + g are the inputs and a frame i + k with $0 < k < g$ is the target, with $t = k/g$. Mixing the three gaps during training lets one model serve all evaluation settings. The loss is the L1 distance between Laplacian pyramids of the prediction and the target, as used in the original RIFE work [1], which penalises errors at several spatial scales:

$$L = \Sigma_l \, \| \, \Lambda^l(\hat{I}_t) - \Lambda^l(I_t) \, \|_1 \tag{3}$$

where Λ^l is level l of the Laplacian pyramid. Augmentation includes random horizontal and vertical flips, 90-degree rotations and temporal reversal, in which I_0 and I_1 are swapped and t is replaced by $1 - t$. The model is trained with the AdamW optimiser, a batch size of 8 and a cosine learning rate schedule, on free GPU sessions from Google Colab and Kaggle (NVIDIA T4 and P100, 16 GB). The checkpoint with the lowest validation loss is kept.

3.5 Baseline Methods

Three baselines are implemented. **Linear blending** computes $\hat{I}_t = (1 - t)\, I_0 + t\, I_1$. **Farneback interpolation** computes forward and backward flow with the OpenCV implementation (pyramid scale 0.5, 3 levels, window size 15, 3 iterations, polynomial neighbourhood 5, sigma 1.2), approximates the intermediate flows with the linear model used in Super SloMo [2], warps both inputs with cv2.remap and blends them with weights $(1 - t)$ and t. **Super SloMo** [2] is fine-tuned from its public checkpoint on the same SEVIR training split and with the same augmentation as RIFE, so the comparison between the two learned models is fair.

3.6 Web Dashboard

The dashboard lets a user choose an event and an input gap, run interpolation, and play the resulting sequence. Real and generated frames are shown side by side, and every generated frame carries a visible GENERATED label so that it is never mistaken for an observation. A difference map between the generated frame and the held-out real frame can be toggled. Per-frame PSNR, SSIM and MAE values are drawn as line charts, and a summary table compares all four methods. MongoDB stores one document per frame with the event identifier, timestamp, source, method, file path and scores; the images themselves are stored as PNG files and served by Express.

4. EXPERIMENTAL SETUP

4.1 Evaluation Protocol

All methods are evaluated on the SEVIR test events. For each gap $g \in \{2, 4, 6\}$ frames, every valid window of the test sequences provides the two input frames and $g - 1$ real frames in between, which are held out and used only as ground truth. For the 10-minute gap one frame is generated per window, for the 20-minute gap three, and for the 30-minute gap five. The INSAT-3D data has no 5-minute ground truth, so it is used for visual inspection only, and to check generated frames against the real INSAT-3D frame that falls in the middle of a 60-minute window.

4.2 Evaluation Metrics

Three metrics are reported on the normalised [0, 1] frames. The mean absolute error over N pixels is:

$$MAE = (1/N)\, \Sigma_i \, | \, \hat{I}_t(i) - I_t(i) \, | \tag{4}$$

The peak signal-to-noise ratio, with peak value 1 and mean squared error MSE, is:

$$PSNR = 10 \log_{10} (1 / MSE) \quad (5)$$

The structural similarity index (SSIM) [6] compares local luminance, contrast and structure of the two images, which is closer to how a person judges cloud shape:

$$SSIM = A / B \quad (6)$$

with $A = (2\mu_x\mu_y + c_1)(2\sigma_{xy} + c_2)$ and $B = (\mu_x^2 + \mu_y^2 + c_1)(\sigma_x^2 + \sigma_y^2 + c_2)$, where μ, σ^2 and σ_{xy} are local means, variances and covariance computed in an 11 × 11 Gaussian window, and c_1, c_2 are small constants that keep the division stable.

MAE is also converted back to kelvin to give an error in brightness temperature, which is easier for meteorologists to interpret. Inference time per generated frame is measured on the same GPU for all methods.

4.3 Implementation Details

The interpolation code uses Python 3.10, PyTorch 2.x, OpenCV 4.x, NumPy, h5py and netCDF4, and metrics are computed with scikit-image. The web application uses Node.js 20, Express 4, MongoDB 7 and React 18. All random seeds are fixed, and the same test windows are used for every method so that the scores are directly comparable.

5. RESULTS AND DISCUSSION

Table 2 gives the average scores of all methods on the SEVIR test set for the three input gaps. Higher PSNR and SSIM and lower MAE indicate better frames.

Table -2: Average scores on the SEVIR test set (PSNR in dB)

Gap	Method	PSNR	SSIM	MAE
10 min	Linear blending	[ ]	[ ]	[ ]
	Farneback	[ ]	[ ]	[ ]
	Super SloMo	[ ]	[ ]	[ ]
	RIFE (fine-tuned)	[ ]	[ ]	[ ]
20 min	Linear blending	[ ]	[ ]	[ ]
	Farneback	[ ]	[ ]	[ ]
	Super SloMo	[ ]	[ ]	[ ]
	RIFE (fine-tuned)	[ ]	[ ]	[ ]
30 min	Linear blending	[ ]	[ ]	[ ]
	Farneback	[ ]	[ ]	[ ]
	Super SloMo	[ ]	[ ]	[ ]
	RIFE (fine-tuned)	[ ]	[ ]	[ ]

[Fill Table 2 with the values obtained from the test runs and describe them here: which method scores best at each gap, how much the scores drop as the gap grows from 10 to 30 minutes, the MAE in kelvin, and the inference time per frame for each method.]

Some behaviour can be checked qualitatively on the dashboard independent of the averages. Linear blending is expected to show double cloud edges whenever a cloud moves more than its own width between the inputs. Flow-based methods avoid this but can smear thin cirrus or stretch clouds at the image border where flow points outside the frame. The fusion mask in RIFE should help in regions where convection develops between the two inputs, since a new cloud top is visible only in I_1. A known limitation of all interpolation methods is that they cannot create events that are absent from both inputs: a cell that forms and decays within one 30-minute gap will not appear in the generated frames. For this reason the dashboard always labels generated frames and reports their error against real frames wherever ground truth exists.

6. CONCLUSION

This paper presented a software system that increases the temporal resolution of geostationary infrared imagery from 30 minutes to 5 minutes using a RIFE network fine-tuned on satellite frames. The pipeline covers reading NetCDF and HDF5 products from SEVIR, NOAA and MOSDAC, preprocessing, arbitrary-timestep interpolation, evaluation against real held-out frames at 10, 20 and 30-minute gaps, and a MERN dashboard that separates real and generated frames clearly. Because no new satellite hardware is needed, the approach can be applied to archived data as well as to near real-time feeds. Future work includes multi-band input combining water vapour and window channels, training directly on INSAT-3DR and INSAT-3DS rapid-scan data, adding physical constraints such as advection consistency to the loss, and uncertainty maps that show where the generated frames are least reliable.

ACKNOWLEDGEMENT

We thank our guide, Prof. [Guide Name], for guidance throughout this work, and the Department of Computer Engineering, [College Name], for providing facilities. We also acknowledge the SEVIR team, NOAA and MOSDAC (ISRO) for making their satellite data publicly available.

REFERENCES

1. Z. Huang, T. Zhang, W. Heng, B. Shi and S. Zhou, “Real-Time Intermediate Flow Estimation for Video Frame Interpolation,” in Proc. European Conference on Computer Vision (ECCV), 2022, pp. 624–642.
2. H. Jiang, D. Sun, V. Jampani, M.-H. Yang, E. Learned-Miller and J. Kautz, “Super SloMo: High Quality Estimation of Multiple Intermediate Frames for Video Interpolation,” in Proc. IEEE/CVF CVPR, 2018, pp. 9000–9008.
3. M. S. Veillette, S. Samsi and C. J. Mattioli, “SEVIR: A Storm Event Imagery Dataset for Deep Learning Applications in Radar and Satellite Meteorology,” in Advances in Neural Information Processing Systems (NeurIPS), vol. 33, 2020, pp. 22009–22019.
4. G. Farnebäck, “Two-Frame Motion Estimation Based on Polynomial Expansion,” in Proc. Scandinavian Conference on Image Analysis, LNCS vol. 2749, 2003, pp. 363–370.
5. B. K. P. Horn and B. G. Schunck, “Determining Optical Flow,” Artificial Intelligence, vol. 17, pp. 185–203, 1981.
6. Z. Wang, A. C. Bovik, H. R. Sheikh and E. P. Simoncelli, “Image Quality Assessment: From Error Visibility to Structural Similarity,” IEEE Transactions on Image Processing, vol. 13, no. 4, pp. 600–612, 2004.
7. T. J. Schmit, P. Griffith, M. M. Gunshor, J. M. Daniels, S. J. Goodman and W. J. Lebair, “A Closer Look at the ABI on the GOES-R Series,” Bulletin of the American Meteorological Society, vol. 98, no. 4, pp. 681–698, 2017.
8. A. Dosovitskiy et al., “FlowNet: Learning Optical Flow with Convolutional Networks,” in Proc. IEEE ICCV, 2015, pp. 2758–2766.
9. D. Sun, X. Yang, M.-Y. Liu and J. Kautz, “PWC-Net: CNNs for Optical Flow Using Pyramid, Warping, and Cost Volume,” in Proc. IEEE/CVF CVPR, 2018, pp. 8934–8943.
10. Z. Teed and J. Deng, “RAFT: Recurrent All-Pairs Field Transforms for Optical Flow,” in Proc. ECCV, 2020, pp. 402–419.
11. S. Niklaus, L. Mai and F. Liu, “Video Frame Interpolation via Adaptive Separable Convolution,” in Proc. IEEE ICCV, 2017, pp. 261–270.
12. W. Bao, W.-S. Lai, C. Ma, X. Zhang, Z. Gao and M.-H. Yang, “Depth-Aware Video Frame Interpolation,” in Proc. IEEE/CVF CVPR, 2019, pp. 3703–3712.
13. T. J. Vandal and R. R. Nemani, “Temporal Interpolation of Geostationary Satellite Imagery With Optical Flow,” IEEE Transactions on Neural Networks and Learning Systems, vol. 34, no. 8, pp. 3245–3254, 2023.

14. Meteorological and Oceanographic Satellite Data Archival Centre (MOSDAC), Space Applications Centre, ISRO. [Online]. Available: https://www.mosdac.gov.in
15. A. Paszke et al., "PyTorch: An Imperative Style, High-Performance Deep Learning Library," in Advances in Neural Information Processing Systems (NeurIPS), vol. 32, 2019, pp. 8024–8035.
16. X. Shi, Z. Chen, H. Wang, D.-Y. Yeung, W.-K. Wong and W.-C. Woo, "Convolutional LSTM Network: A Machine Learning Approach for Precipitation Nowcasting," in Advances in Neural Information Processing Systems (NeurIPS), vol. 28, 2015, pp. 802–810.
17. O. Ronneberger, P. Fischer and T. Brox, "U-Net: Convolutional Networks for Biomedical Image Segmentation," in Proc. MICCAI, LNCS vol. 9351, 2015, pp. 234–241.