ISSN: 2093-5374 / E-ISSN: 2093-5382

Video Stabilization by a Hybrid Structure of CNN: RNN for a Video Surveillance System

C.K. Siva Ranjani[1*], Dr.V. Vallinayagam[2], Dr.P. Gururama Senthilvel[3], Dr.M. Shakila[4], Dr.B. Abirami[5], and Dr.J. Nithisha[6]

[1*]Department of Computer Science and Engineering, Saveetha School of Engineering, Saveetha Institute of Medical and Technical Sciences, Saveetha University, Chennai, Tamil Nadu, India. sivaranjani9017.sse@saveetha.com, https://orcid.org/0000-0002-6782-179X

[2]Professor, Department of Mathematics, St. Joseph's College of Engineering, Chennai, Tamil Nadu, India. vngam19@gmail.com, https://orcid.org/0000-0001-6715-6227

[3]Professor, Department of Computer Science and Engineering, Saveetha School of Engineering, Saveetha Institute of Medical and Technical Sciences, Saveetha University, Chennai, Tamil Nadu, India. gurupandian.cse@gmail.com, https://orcid.org/0000-0001-8666-1544

[4]Department of Computer Science Engineering, Saveetha School of Engineering, Saveetha Institute of Medical and Technical Sciences, Saveetha University, Chennai, Tamil Nadu, India. shakilam.sse@saveetha.com, https://orcid.org/0009-0001-4051-5768

[5]Assistant Professor, Department of Computer Science and Engineering, Saveetha School of Engineering, Saveetha Institute of Medical and Technical Sciences, Saveetha University, Chennai, Tamil Nadu, India. abivsb.sanrak@gmail.com, https://orcid.org/0009-0001-9187-9238

[6]Associate Professor, Department of Computer Science and Engineering, Saveetha School of Engineering, Saveetha Institute of Medical and Technical Sciences, Chennai, Tamil Nadu, India. nithisha.j@gmail.com, https://orcid.org/0009-0008-1055-0686

Received: October 18, 2025; Revised: December 13, 2025; Accepted: January 21, 2026; Published: March 31, 2026

Abstract

Video stabilization is important for enhancing the visual quality of both surveillance and consumer videos by minimizing undesired jitter and motion shake. Older methods are based upon pixel-space optimization, motion heuristics, or huge training sets, which tend to have non-convex optimization challenges and require an accurate optical flow estimate. The study proposes a hybrid CNN-RNN model, in which video stabilization is achieved by optimizing the CNN parameter space and refining the intra-frame hidden states via an RNN block. In contrast to traditional approaches that rely on learning, our model is directly trained on a specific input video, and it will overfit its parameters to obtain video-specific optimization. The CNN is a differentiable optimizer in a high-dimensional parameter space, and the RNN based on ConvLSTM makes use of inter-frame recurrence and intra-frame recurrence to enhance temporal consistency without adding extra architectural elements. As experimental results on DeepStab and NUS data show, our algorithm has a better PSNR, better SSIM, better ITF, and much faster execution time than the latest methods do. The suggested method is efficient, strong against parallax and dynamic images, and applicable in real time to the implementation of video surveillance systems.

Keywords: Video Stabilization, CNN, RNN, Video Surveillance, Video Processing.

Journal of Wireless Mobile Networks, Ubiquitous Computing, and Dependable Applications (JoWUA), volume: 17, number: 1 (March - 2026), pp. 648-668. DOI: *10.58346/JOWUA.2026.I1.036*

*Corresponding author: Department of Computer Science and Engineering, Saveetha School of Engineering, Saveetha Institute of Medical and Technical Sciences, Saveetha University, Chennai, Tamil Nadu, India.

1 Introduction

The high growth rate of surveillance cameras in public, commercial, and domestic settings has also generated high demand for efficient video analytics machines. In recent surveillance systems, automated processing is essential in the detection of abnormal events, tracking of objects, and analyzing behavior (Petrushin, 2005; Popoola & Wang, 2012; Remagnino et al., 2011). The functionality of these downstream applications is, however, often undermined by the occurrence of unwanted camera motion, jitter, and motion blur, which is often a result of handheld recording, external interference, low-quality sensors, or hardware restrictions. These degradations not only decrease the visual quality but also the accuracy of the high-level machine vision algorithms (Jelena & Srđan, 2023; Arora, 2024).

The digital video stabilization is an attempt to have the undesirable camera motion compensated by estimating the inter-frame transformations, smoothing motion trajectories, and creating geometrically consistent frames. Traditional stabilization pipelines will use three sequential processes: (i) motion estimation, (ii) motion smoothing, and (iii) frame warping. Although they are widely used, some of the traditional algorithms make limiting assumptions like planar homography, motion on a global scale, or trustworthy tracking of features (Gleicher & Liu, 2008; Buehler et al., 2001; Jeevanand et al., 2014). These assumptions often fail in real-world surveillance evidence, which might contain parallax, occlusions, depth variations, high-speed rotations, and low-texture areas. Recent efforts in deep learning-based stabilization have demonstrated good results through the learned motion priors being directly learned using data. However, the vast majority of currently available methods rely on extensive annotated data, computationally expensive structures, or pixel-based optimization processes that do not scale well with video resolution and length. Additionally, CNN-based models generally do not have temporal memory, whereas recurrent models can encode temporal correlations but fail to model intra-frame refinement expressly, which prevents them from recovering fine-grained information on motions in difficult scenes (Kadhim et al., 2023; Wieschollek et al., 2017; Madhan & Shanmugapriya, 2024).

In order to address these shortcomings, a hybrid CNN-RNN model is suggested, and the video stabilization issue is formulated as a parameter-space optimization problem as opposed to an estimation in pixel space. The main incentive of this design is that the CNN parameter space is a smoother and more structured space than raw pixel motion fields, and thus it can be optimized using a high-dimensional method. A convolutional neural network is trained on the input video instead of estimating the number of motion vectors per video, which are millions. It is a process of overfitting that is designed and controlled purposefully and based on the notion of the so-called deep prior. This allows the network to be a potent differentiable optimizer, generating spatially coherent warp fields without the need to have a large training corpus.

In order to complement the CNN part, a recurrent framework is employed based on ConvLSTM, which promotes the identification of the temporal connections between the frames. Unlike in traditional RNN models, in which the hidden states are updated after each frame, our model proposes the use of intra-frame iteration, in which the hidden state is updated after refinement with the same RNN parameters. This enables the network to stabilize the motion paths in a better way without compromising the architectural simplicity and computational efficiency (Soorya et al., 2017; Shnayderman et al., 2006). The inter-frame and intra-frame repetition of the model allows it to spread contextual information both through time and iteration, leading to a better resistance to blur, parallax, and even dynamic scene content. The proposed hybrid CNNRNN model therefore offers a single formulation of motion estimation, temporal refinement, and warp-field optimization specific to the nature of each particular

video (Matsushita et al., 2006; Liu et al., 2023). Extensive experiments of DeepStab and NUS have shown that our algorithm has a high PSNR, SSIM, ITF, and temporal smoothness in comparison to classical pipelines of stabilization and new deep learning methods (Bhat et al., 2007; Liu et al., 2012; Camgözlü & Kutlu, 2023; Su et al., 2017).

The model is also real-time, and hence it is very applicable in the real-world implementation of surveillance systems and embedded platforms (Hyun Kim et al., 2017; Ulyanov et al., 2018). To conclude, the contributions of this work are as follows:

- CNN performs parameter-space optimization to generate smooth warp fields. RNN provides temporal coherence via inter-frame and intra-frame recurrences.
- Multiple hidden-state refinements per frame yield significantly better deblurring and stabilization, with no extra parameters.
- The optimization of CNN weights is an alternative to solving million-way pixel-level motion vectors.
- Incorporates locally adaptive penalties to handle parallax and unreliable motion regions.
- Extensive evaluation showing that the method achieves higher PSNR, SSIM, ITF, and lower distortion than state-of-the-art methods while running in real-time.

The following is the outline of this article: The second section provides an overview of the literature on video stabilization, RNNs, and CNN models. In Section 3, the hybrid CNN-RNN architecture is proposed, and the functionality of the architecture is explained. The experimental findings on the proposed approach and comparison with current methods for video stabilization are presented in Section 4. Section 5 concludes by summarizing the findings of the proposed model.

2 Related Work

Here, a literature review of the available sources on the use of neural networks to develop video stabilization, video deblurring, and similar topics is provided.

2.1 Video Deblurring

Prior research on video deblurring depended on the concept of lucky imaging, in which clear contents were substituted for fuzzy equivalents at the pixel (Matsushita et al., 2006) and patches (Cho et al., 2012) levels. Kernels are inferred from inter-frame relations in deconvolution-based algorithms, which were later the subject of much study. The use of time-based data allowed for the prediction of global motion and the creation of a clear panoramic image from a hazy video (Li et al., 2010). As a means of dealing with various blurred areas, (Wulff & Black, 2014). developed layered blur models, which include layer segmentation and deconvolution of individual layers to enhance blur kernel and latent image estimation. Using bidirectional optical flows to mimic locally variable blur kernels, (Kim et al., 2017). presented a segmentation-free technique for dynamic video deblurring. The local blur kernels and latent images are variables in this framework, which presents the task as a non-convex energy minimization problem. To get around this, several deblurring deconvolution methods (Kim et al., 2017; Hyun Kim et al., 2013) Optimize the energy function repeatedly.

2.2 RNN+CNN

Both spatial (via CNNs) and temporal (by LSTMs) features may be effectively learned by deep learning systems on their own. Learning elements pertaining to spatial and temporal relations constitute what are known as spatiotemporal networks (STNs) (Zhang et al., 2018). In STNs, spatiotemporal characteristics are extracted using a combination of CNNs and LSTMs (Chalapathy & Chawla, 2019). Once CNN has processed the data, the next LSTM will take its cues from the CNN structure's output (which may be ResNet or AlexNet, for example). In order to identify suspicious occurrences in video datasets, several researchers have begun to use detection methods such as (Liu et al., 2013). In addition, a new method has evolved whereby a convolutional layer filters the CNN output before it enters the LSTM structure. (Medel & Savakis, 2016). Convolutional Long Short-Term Memory (ConvLSTM) describes this novel method. With a convolutional layer, the number of parameters is significantly reduced, as opposed to being completely connected in an LSTM. Consequently, the likelihood of overfitting is reduced, and the model's performance may be improved.

2.3 Video Stabilization

In general, 2D approaches are efficient and have minimal computing complexity. A number of issues may arise with 2D approaches. One issue is that monitored 2D characteristics aren't always accurate because of things like lighting changes and motion blur. It is also challenging to get lengthy feature tracks in videos with a lot of occlusions. They (Yu & Ramamoorthi, 2018) follow the two-dimensional feature points and finds a grid that covers them, making them smooth. To begin the algorithm, they (Grundmann et al., 2011) needs a camera path determined from feature tracks. In (Sultani et al., 2018) They require basic motion situations and lengthy feature tracks as well. The relative locations of feature points are the focus of several approaches. Stabilize the eigen-trajectories that have been derived in (Liu et al., 2011). Use epipolar geometry to keep feature points in their relative positions was proposed by Goldstein & Fattal, (2012). Try to maintain the relative locations of feature points as well. The quality of the recorded characteristics still determines how well these works operate.

Secondly, parallax-effect movies are notoriously difficult to stabilize using a parameterized motion model alone, as pixel movements within the same frame aren't always constrained by a similar homography constraint. The video is stabilized using a full-frame homography in Matsushita et al., (2006), which treat the scene as a plane. Both (Liu et al., 2014; Yadav et al., 2024) use grids and local homography to split the frames, but they aren't good at dealing with complicated depth fluctuation. In (Gibson & Salamonson, 2023), they distort the first frames by means of optical flow. On the other hand, motion discontinuities have a significant impact on the pixel profile provided in Goldstein & Fattal, (2012). Because of this, heuristics for determining foreground and background information are still required, as is meticulous inpainting of areas where motion differs from the backdrop.

When compared to generic 2D approaches, ours is superior for feature tracking due to its ability to withstand local optical flow defects and track all pixels. To avoid having to recreate the scene's 3D structure, a non-parametric frame warping method is turned on to solve the parallax effects. In spite of the fact that optical flow is applied to produce stabilized frames, like the technique in, do things rather differently. Unlike the pixel profile, which only records the motion vectors at each pixel location, our method really tracks the motion of each pixel. This means that our strategy is parallax resilient and doesn't need filling in the zones of motion discontinuities. The large-scale nonconvex issue that emerges from our formulation, however, defies easy quadratic form, as shown in (Goldstein & Fattal, 2012).

3 Proposed Work

The proposed video stabilization framework integrates a hybrid CNN–RNN architecture designed to jointly model temporal dynamics and spatial motion correction through parameter-space optimization. As illustrated in the revised figure 1, the overall pipeline consists of two main components: (i) a recurrent module that performs inter-frame and intra-frame refinement of latent representations, and (ii) a CNN-based optimizer that estimates smooth, spatially coherent warp fields from optical-flow inputs. In the first stage, sequential video frames are processed by a ConvLSTM recurrent unit that maintains a hidden state capturing temporal structure across frames. Instead of having the hidden state updated on a per-timestep basis, unlike normal RNNs, an intra-frame iteration mechanism is suggested in terms of which the hidden state is iteratively optimised on several occasions based on the use of a single-cell or dual-cell architecture. In the single-cell scheme, a particular recurrent unit is shared by the initial and iterative updates, and therefore, the scheme becomes efficient in parameters; in the dual-cell scheme, different cells are used to independently predict the latent frames and model the hidden-state refinement, and thus, the scheme is more expressive, but it consumes more parameters. This recurring iterative process allows deepening time integration and minimizing the cumulative effect of residual blur by changing the network architecture. The trained latent image output of the RNN is then a latent-to-sample image that is passed to the CNN optimization network that produces a dense pixel-wise warp field to stabilize each frame. In stabilization, rather than directly optimizing the millions of pixel motions as with classic algorithms, stabilization is stated as a CNN parameter-space optimization problem, which is a combined smoothness and structural prior of convolutional filters. The CNN is fed initial optical flow fields and is trained on the target video, whereby the intention is to overfit it to create a warp field that fits the sequence. The optimization problem incorporates 3 terms: (i) a motion-consistency loss, which makes the warped frames align by minimizing Euclidean distance, (ii) a regularization term, which makes the warp field smooth and (iii) an adaptive weighting scheme, which heightens the regularity of warp fields in areas with high magnitude of the motion vectors in order to eliminate unreliable or noisy motion estimates. This combination of RNN-based temporal recurrence and CNN-based parameter-space optimization allows the system to overcome non-convexity challenges inherent in pixel-level motion estimation, while maintaining robustness to parallax, dynamic objects, and local flow artifacts. The resulting framework provides a unified, end-to-end approach for stabilizing complex real-world video sequences. The next section shows the empirical considerations that can prove the usefulness of this hybrid design on a variety of datasets and benchmarks.

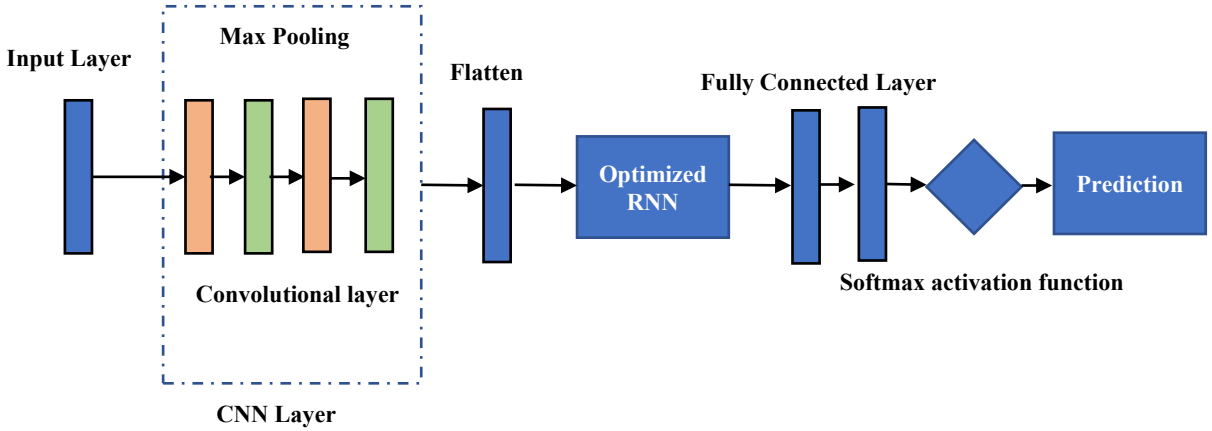

Figure 1: Proposed model architecture using CNN-RNN framework

3.1 RNN Model

Since CNNs do not have any temporal connections, the hidden state that RNNs use to their advantage is the most important aspect when competing with CNNs. Having strong hidden states is crucial for this reason, as they will aid in making more accurate output predictions both at the present and upcoming frames. So, before passing it on to the next RNN cell, try to make more use of hidden states using intra-frame iteration. Using our basic RNN cell design, and put this theory into action. Beginning with the blurry input B_t and the prior hidden state $h_t - 1$, use our RNN cell to calculate the first hidden state $\widehat{h_t^0}$ at an appropriate step t. In order to update the concealed state, repeat the process of feeding $\widehat{h_t^0}$ back to the cell without altering B_t. Lastly, at the time step, update the hidden state $\widehat{h_N^0}$ and construct an unseen output frame L_t after N repetitions of updating the hidden state. Despite the iterations, the blur feature extraction tool F_B along with the hidden frame predictor F_L are only utilized once.

Two distinct iteration methods are shown here: the single cell approach and the dual cell method. When estimating the initial and updated hidden states using the single cell technique, keep the parameters constant. In contrast, a dual cell approach makes use of two RNN cells, one of which serves a specific function. Predicting latent frames and updating hidden states only utilize the second cell. While there are more parameters needed for the dual cell technique compared to the single cell approach, the performance advantage may be substantial since separate sets of parameters can be assigned various functions. Going forward, use the prefix C1 to indicate single-cell models and C2 to identify dual-cell models. Finally, append the iterations of the hidden state with the suffix H. In the dual cell model, the symbol C2H2 indicates that the hidden state is updated twice. Here the detail Algorithm 1's two techniques for updating hidden states intra-frame. From an architectural point of view, our techniques effectively deepen RNN cells, which increases their receptive fields and capacities.

The fuzzy video can be represented as $B = \{B_t\}$, the ground-truth sharp video as $S = \{S_t\}$, and the predicted hidden video as $L = \{L_t\}$, where t is a frame index in the interval $\{1 \ldots T\}$. In order for temporal information to flow across video frames, and build our basic structure as a recurrent neural network, similar to Bhat et al., (2007). Next, our network applies recurrence operation on the input video, which is blur video.

$$(L_t, h_t) = F(B_t, h_{t-1})$$

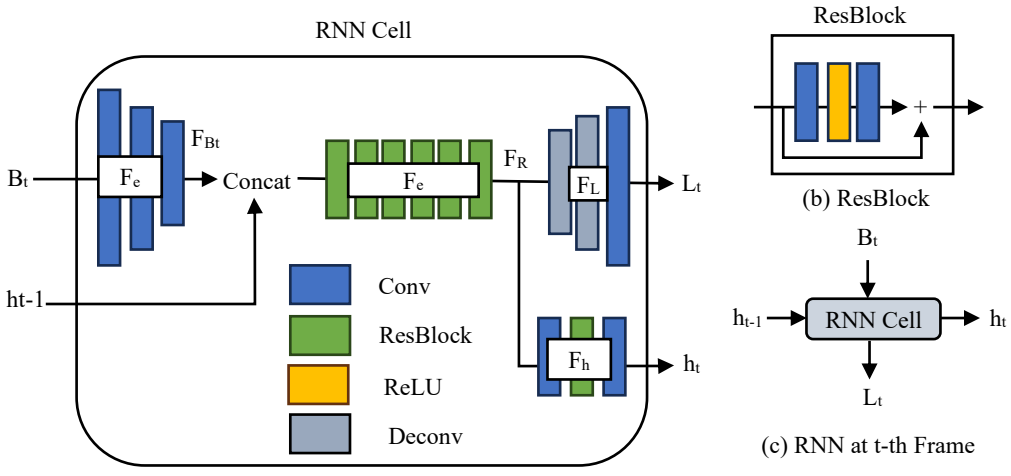

Figure 2: RNN model for intra-frame iteration

Here, our RNN cell is denoted by F. The cell is made up of many parts, including $F_B, F_R, F_L, and\ F_h$, as seen in figure 2. The feature f_{B_t} is first extracted from a fuzzy frame via F_B. Afterwards, F_L, and F_h calculate the latent frame L_t with hidden state h_t, respectively, using the intermediate feature f_{B_t} that is produced by F_R. In the t-th time-step, the hidden state ht is created and will be transmitted to the $t + 1$-th time-step. The initial value of h_0 is set to zero.

3.2 CNN Model

An easy way to simulate the transition in appearance is to determine the initial optical flow between successive video frames and then locate transformations between frames that will reduce pixel motion. The result obtained from this RNN phase is used as input in the rest of the study, which then covers optimization to stabilize the video that results. Reducing the $x\ and\ y$ coordinates of each pixel to a single pixel ID (i) simplifies the notation. F_t is the two-channel image that originally represented the optical flow from t to $t + 1$; it encoded the movement of all the pixels from frame I_t to frame I_{t+1} . Take the example of pixel $i's$ location in frame I_t, which may be represented as $p_{i,t}$. The image that corresponds to it in frame I_{t+1} may be described as equation (1)

$$P_j, t + 1 = P_{i,t} + F_t(P_i, t) \tag{1}$$

It is also possible to calculate a backward optical flow, which maps the pixels in the image to each other. Adding $_{it}$ to the frame I_{t+1} is given in equation (2)

$$P_{l,t} = P_{k,t+1} + \bar{F}_t\ (P_k, t + 1\ ) \tag{2}$$

The warping of every original frame stabilizes the output frames. The warping process is an affine transformation H_t of the two-dimensional image and a per-pixel warp field W_t.

Hence, frame I_t the pixel i and l, the distorted pixels are depicted by using the below equation (3)

$$\widehat{P_{i,t}} = H_t P_{i,t}^h + W_t\left(P_{i,t}\right)$$
$$\widehat{P_{l,t}} = H_t P_{l,t}^h + W_t(P_{l,t}) \tag{3}$$

Where the uniform visualization of $P_{i,t}$ and $P_{l,t}$ are denoted by $P_{i,t}^h$ and $P_{l,t}^h$, respectively. Similarly, its distorted image matching I_{t+1} is given by equation (4)

$$\widehat{P_{J,t+1}} = H_{t+1} P_{j,t+1}^h + W_{t+1}(P_{j,t+1})$$
$$\widehat{P_{k,t+1}} = H_{t+1} P_{k,t+1}^h + W_{t+1}(P_{k,t+1}) \tag{4}$$

Minimizing the distance between the warped pixel locations, as measured by the Euclidean distance, is the objective given in equation (5)

$$E_0(W, H) = \frac{1}{wh(T-1)} \sum_{t=1}^{T-1} \left(\left(\sum_{i=1}^{wh} \left\| \widehat{P_{i,t}} - \widehat{P_{J,t+1}} \right\| \right)^2 + \left(\sum_{i=1}^{wh} \left\| \widehat{P_{i,t}} - \widehat{P_{k,t+1}} \right\| \right)^2 \right) \tag{5}$$

This is equal to T frames multiplied by wh, the total number of pixels in a frame. You just need to average over i and k since the mapping from $p_{i/l,t}$ to $p_{i/l,t+1}$ is 1-to-1.

Because scenes are so complicated, the original visual flow might not have been right in some places. In addition, the camera's movement may not affect the scene's items. Introducing artifacts might be a result of blindly optimizing objective function (5).

The error function is given by, equation (6)

$$E_r(W) = W_t - D(D^T D)^{-1} D^T w_t \quad (6)$$

To ensure that the output warp field is near to a linear warp field, employ this error as a restriction. It should be noted that by adjusting D, can manage the linear warping restriction at the pixel level using this formulation. In our experiment, for instance, use a 20x20 grid to cover each frame and fill D using the weight of every image pixel in its neighbouring grid cell.

In addition, areas with significant movements make the initial optical flow F_t less dependable. Keeping this in mind, usually get W_t with fewer discontinuities by increasing the regularization value (6) for big motion areas, and trust the optical flow and lower it for tiny motion regions. Pixel motion derived from the initial optical flow may be used to estimate the motion scale measurement by equation (7)

$$E_p = F_t^2 + \bar{F}_t^2 \quad (7)$$

By integrating equations (5), (6), and (7), can express our optimization problem as equation (8)

$$\min_{W,H} E_0(W,H) + \lambda \left\| E_p \, . E_r(W) \right\|_1 \quad (8)$$

Here, the hyperparameter $\boldsymbol{\lambda}$ controls the overall level of regularization. Apply it as a pixel-wise weight to E_r because the original optical flow magnitude, E_p, is equal to the size of the regularization, E_r. It should be noted that the L_1 norm is used for this regularization in order to promote sparsity in the warp field and prevent excessive compensation to incorrect areas in the initial optical flow.

3.2.1 CNN Optimization Model

Make note that the variables W_t and H_t, which represent the optical flow field and the 2D affine transformation, respectively, for each frame, are the unknowns in equation (8). At the industry standard 480p resolution, there are almost 123 million unknown motion vectors in a 300-frame video clip. Because of the memory and computing requirements, it is usually not feasible to directly optimize a problem of this magnitude. In addition, there are a lot of local minima in the complicated high-dimensional energy landscape, making optimization a challenge. Instead of looking in the issue space for a solution, propose to look in the parameter space of the CNN neural network. Actually, the CNN neural network is serving as our optimizer. Learning on huge datasets has always been done differently than our strategy. Because the objective function (8) is applied immediately to the input video, the network weights are optimized without a training set. An optical flow-based robust stabilizing formulation can be applied at once since this non-convex multidimensional optimization problem becomes feasible when a network is used.

Our approach is the first study that is aware of to apply this concept to video stabilization problems. The optical flow field is spatially smooth for real-world situations, even if it is represented pixel-wise. Hence, a parameterized function adequately describes our warp field $\{W, H\}$. It is necessary to create a sophisticated and distinguishable parameterized function due to the complexity of this procedure. Instead of creating a convolutional neural network manually, the prefabricated, optimal solution to this task is selected. $G(\theta)$ is the function that denotes the neural network, with θ standing for the network's parameters. To get the required warp field $\{W, H\} = G(\theta)$, the network has to have a set of weights found that will allow it to generate the desired output. So, to rephrase the optimization problem (8) given the final optimization in equation (9).

$$\min_{\theta} E_0(G(\theta)) + \lambda \left\| E_p \, . E_r(G(\theta)) \right\|_1 \quad (9)$$

Finding the parameters θ becomes the objective when the network is trained on a single footage clip. This means that (9) is distinguishable with regard to the parameters of the network, since our goal function is comprised of basic linear and quadratic functions of $\{W\ and\ H\}$.

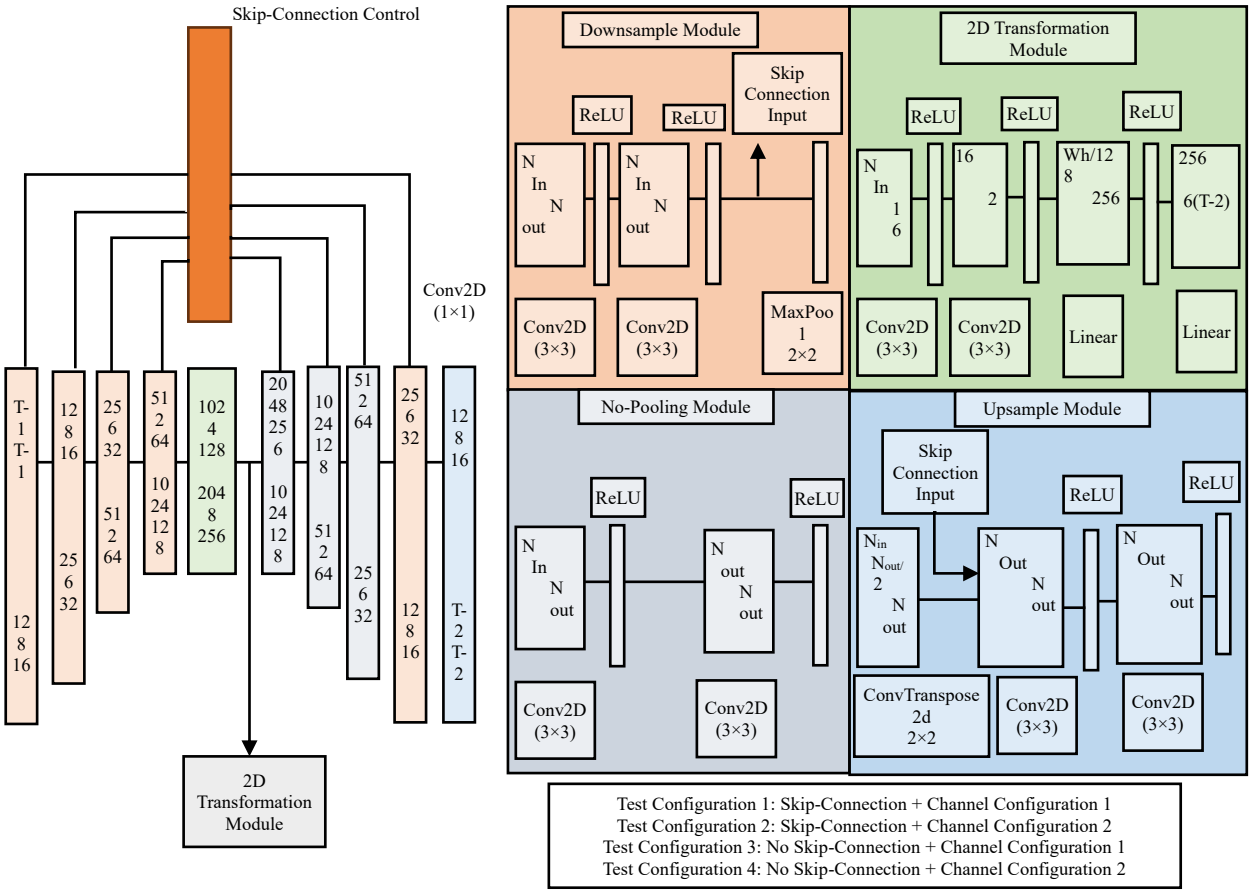

Figure 3: CNN model for optimization

Figure 3 displays our network architecture. A collection of $T-1$, the primary optical flow fields F, calculated from input video frames, constitutes the input of the network. Multiple channels are used to transmit the optical flow field frames. Five levels of downsample module encoding the input. With the exception of the final module, which only increases the number of channels, all of the downsample modules reduce the frame size by 2. Starting with four layers of upsample modules, the decoder continues with a $1-1$ kernel-size output convolutional layer. The decoder produces the necessity of the warp field W required. The encoded data is also processed by a 2D transformation module, which has 2 linear layers and 2 convolutional layers. The aim of this module is to generate the $2D$ affine transform matrix H. The output $\{W, H\}$ pairs with $T-2$. It is intended to optimize the overfitting of the single input video, to optimize the network parameters with the optical flow of the video only. In order to do this, normalization of learning on the network weights and dropout layers would be discouraged. It seems that the U-Net structure provided the inspiration for the figure 3 modular deep learning architecture, which was created for image modification or segmentation tasks. The encoder (on the left), the transformation module (in the middle), and the decoder (on the right) are the three primary parts of the design. Multiple convolutional layers make up the encoder, and they downsample features while gradually extracting them. The orange colour is used to denote these layers. In order to prevent the loss of fine-grained spatial information during downsampling, the model establishes skip links when features

are retrieved from each encoder block to its associated decoder block. The green-highlighted 2D Transformation Module is crucial to the system; it compresses the input into a feature representation by processing the deeply encoded features with additional convolutional layers and linear transformations.

In order to restore the image to its original proportions, the architecture on the decoder side (blue and purple) uses either upsampling or no-pooling techniques. Combining the encoder's skip-connected features with higher-resolution feature maps obtained by transposed convolutions, the upsampling block assists in accurate reconstruction. There are two test setups that stand out at the bottom of the image. The first test setup involves downsampling and transformation with skip connections and certain channel combinations. Second Test Configuration places an emphasis on exclusively learnt representations in situations when skip connections are not present. The core operations of each modular block, including Conv2D, ReLU activations, MaxPooling, and upsampling, are described on the right side. It is possible to optimise performance for different visual tasks by experimenting with alternative combinations of pooling, upsampling, and transformation layers, due to the architecture's overall flexibility.

Algorithm:1-CNN-RNN Model for Video Stabilization

Input: Blur Video Footage

Output: Stabilized Video

1 $Process\ Each\ frame\ to\ obtain\ the\ Frame-to-Frame\ Transformation\ \Delta_{Original}$

2: $Using\ \Delta_{Original}, Computes\ blur$

3: $While\ for\ all\ frames\ do$

4: $if\ current\ Iteration\ <\ end\_iter$

5: $Single\ Cell\ Method\ Deblurring/double\ cell\ deblurring$

6.: $Optimizing\ the\ objective\ function\ (8)\ by\ the\ CNN\ model$

7: end

8: return, Stabilized video

9: End while

4 Experimental Design and Results

4.1 Feature Extraction Comparison

The DeepStab dataset (Nah et al., 2017) was used in this research in order to measure the performance of the proposed video stabilization method. This data will have pairs of both shaky and stable video synchronized; wherein direct, frame-by-frame comparisons could be made during training and testing. Its ground truth of high quality allowed its successful supervision and correct gauge of its stabilization improvements based on conventional measures like PSNR and SSIM. The various patterns of motion and real-world scenarios simulated by DeepStab thus made it quite acceptable to test the robustness and generalization capability of the proposed model.

The video dataset includes five different video clips that may be used to test the video stabilization algorithms in different real-life conditions (Table 1). The initial video (8.avi) has mild jitter and it is captured in a controlled indoor or stable outdoor setting, with the help of handheld and stable camera that offers a comparatively stable background to test the performance of the algorithms. In the second

video (24.avi), the moderate jitter is presented with flowing motion at motion with dynamic environment like walking or by a slow-moving vehicle that creates the natural movement complexity. The third video (38.avi) is relatively shaky during a combination of dynamic and static outdoor shots, which were captured with a handheld camera standing or walking around reflecting the typical user-generated video. The fourth video (42.avi) consists of rolling motion and jitter in an outdoor dynamic environment, emulating recording by a moving individual or a vehicle-mounted camera, and thus it is difficult due to the effects of linear and rotational motion. Lastly, the fifth video (46.avi) is the most extreme one that depicts violent and unstable jitter on high mobility sources such as vehicles or drones in dynamic outdoor settings, which will best test the effectiveness and flexibility of stabilization methods. Together, the dataset gives a high level of variety of motion conditions to construct and test the stabilization models.

Table 1: Input video description

Video File	Resolution	Frame Rate (FPS)	Frame Count	Duration (s)
8.avi	1280×720	59.94	599	9.99
24.avi	1280×720	~30	218	7.27
38.avi	1280×720	29.97	777	25.93
42.avi	1280×720	29.97	967	32.27
46.avi	1280×720	29.97	598	19.95

Our stable video's quality is evaluated by its Peak Signal-to-Noise Ratio (PSNR). The calculated PSNR between two successive frames is given by equation (10):

$$PSNR = 10 \log_{10} \frac{I_{MAX}}{MSE(n)} \tag{10}$$

The PSNR shows a link between the result you prefer and the video you receive. MSE n is the Mean-Square-Error between two frames, and IMAX is the highest pixel number that an image can have. N and M stand for the frame's measurements. An increased PNSR across two stabilized frames means that the video is of good quality. It's given in equation (11).

$$MSE(n) = \frac{1}{MN} \sum_{y=1}^{M} \sum_{x=1}^{N} [I_N(x,y) - I_{n+1}(x,y)]^2 \tag{11}$$

The structured similarity indexing method (SSIM) finds the average number of how similar the two images are in terms of their structure. The SSIM measure compares the brightness, contrast, and structure of a deformed image (x) to an input base image $(y), and$ is given in equation (12).

$$SSIM(x,y) = [l(x,y)]^{\alpha} [c(x,y)]^{\beta} [s(x,y)]^{\gamma} \tag{12}$$

Where $l(x,y), c(x,y)$, and $s(x,y)$ are functions for comparing brightness, contrast, and structure, respectively. $\alpha > 0, \beta > 0$, and $\gamma > 0$ are factors that can be used to change how much each of the three functions has changed.

The downsampled DeepStab dataset is used to test our method and other methods. From table 2, you can see how all of the comparison methods performed in terms of PSNR, SSIM, and running time. Because of these results, it is evident that the CNN-RNN model's intra-frame repetition scheme and the random training method make our model much better than the other best methods. Even though our method has internal repeated processes, it is much faster than the others. The visual deblurring output of the proposed model is illustrated in figure 4.

Shakiness Type	Input Video Frame	Stabilized Video Frame
Mild jitter, mostly stable		
Moderate jitter, smooth motion		
Moderate jitter		
Rolling + jitter		
Aggressive jitter, erratic		

Figure 4: Visual output of our proposed model

Table 2: Performance comparison with existing methods on dataset (Sultani et al., 2018)

Methods	PSNR	SSIM	Speed(fps)
CNN	27.08	0.8429	1.2
Novel RNN	25.19	0.7794	7.37
STRCNN+DTB	26.82	0.8245	9.24
VSRResNet	25.51	0.8834	-
SRGAN	23.58	0.7050	-
Proposed Model	29.32	0.8684	33.67

On the sample in Matsushita et al., (2006). Also, compared how well the CNN-RNN model worked. The training set from Matsushita et al., (2006) to fine-tune our GOPRO models, used a reference source not found. In this case. Compared to models. Matsushita et al., (2006) and Wieschollek et al., (2017). Table 3 shows that our model works better with iterations and regularization.

Table 3: Performance comparison with existing methods on dataset (Matsushita et al., 2006)

Methods	PSNR	SSIM
CNN	30.14	0.8913
Novel RNN	26.98	0.8076
STRCNN+DTB	29.97	0.8696
DBN+HOMOG	27.93	0.9221
DBN+SINGLE	23.63	0 .885
Proposed Model	32.37	0.9865

The Interframe Transformation Fidelity (ITF) rating is based on equation (13)

$$ITF = \frac{1}{N_{frame} - 1} \sum_{k=1}^{N_{frame}-1} PSNR(k) \quad (13)$$

The mean PSNR across two successive frames is called the ITF. Typically, this mean is applied to every statistic in order to provide an approximate estimate for the stabilized video's quality. As with PSNR, higher ITF values reveal a very stable video. The results of the ITF tests for the three video clips are presented in table 4. Based on the results of this analysis, our stabilized videos have a higher ITF than the originals. It is satisfactory that our stabilized movies have an improved ITF.

Table 4: Performance comparison of ITF

Input Video	Original ITF	Stabilized ITF
Video 1 (8.avi)	18.78	22.65
Video 2 (24.avi)	24.21	24.68
Video 3 (38.avi)	25.98	28.65
Video 4 (42.avi)	21.75	26.13
Video 5 (46.avi)	19.02	23.94

4.2 Performance Evaluation of Optical Flow Using the Objective Function

Figure 5 and table 5 shows a quantitative comparison of the input video's quality with that of our result, Liu et al., (2013), Jeevanand et al., (2014) and Kim et al., (2017). An evaluation of the visual change between successive frames in the results is done by calculating the cumulative optical flow across the whole video. Our objective function and its metric are conceptually identical (5). It utilizes the stabilized video to calculate the optical flow. It should be noted that optical flow is frame size standardized; that is, differences between movies with varying frame sizes can be compared. For this statistic, a lower score means higher performance. By comparing each score to that of the input video, the increase in stability could be noticed because much effort was put to ensure that the effect on the overall appearance was minimal (Muralidharan, 2020). As a result, when compared to other ways, ours consistently produces superior results. Take note of the improvement over the pre-stabilized output while using our optimization framework based on CNN-RNN. The obtained result is better than the comparison methods by a percentage in the given datasets, but no clear difference in the quality between the two sets of data can be observed. Further comparisons with different video stabilizing algorithms using the widely-used NUS dataset. are also shown in figure 5. Take an average of these metrics after

randomly selecting 5 videos from each category. This larger dataset for video stabilization yields better results for our technique.

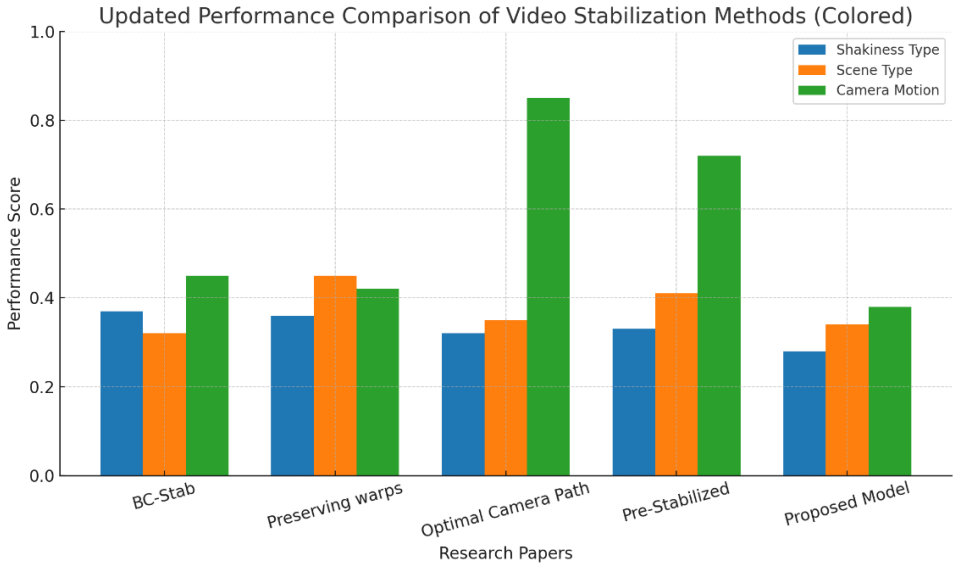

Figure 5: Proposed model performance assessment using different techniques

Table 5: Comparison in de-blurred accuracy and performance with others

Research papers	Shakiness Type	Scene Type	Camera Motion
BC-Stab	0.37	0.32	0.45
Preserving warps	0.36	0.45	0.42
Optimal Camera Path	0.32	0.35	0.85
Pre-Stabilized	0.33	0.41	0.72
Proposed Model	0.28	0.34	0.38

Table 5 is a comparative study of five video stabilization algorithms, namely, BC-Stab, Preserving Warps, Optimal Camera Path, Pre-Stabilized and the Proposed Model, which have been tested by looking at three important criteria of Shakiness Type, Scene Type, and Camera Motion. These metrics have a scale of 0 to 1 with a low value resulting in excellent performance of stabilization. The Proposed Model has the most general performance as it has the lowest shakiness of 0.28, which means that it perfectly reduces visual jitter. It competes also in Scene Type (0.34) and Camera Motion (0.38) indicating the capability to preserve the consistency of the images and minimize unwanted movement between frames. Although Optimal Camera Path has a good score in shakiness (0.32), it has the worst score in camera motion (0.85) indicating that the approach results in drastic change of viewpoint in the effort of stabilizing the video path, which can be distracting to the eye. On the same note pre-stabilized scores moderately in the shakiness (0.33) and marginally better in the scene preservation (0.41), but a high camera motion score (0.72) is observed which is instability in quick scene transitions. In preserving Warps, the scene integrity is more emphasized as its scene score of 0.45 is the highest among them, but this is at the expense of slightly higher shakiness (0.36) and the slightest increase in camera motion (0.42). Lastly, BC-Stab delivers balanced performance across all three metrics, with scores of 0.37 (shakiness), 0.32 (scene), and 0.45 (camera motion), though it does not outperform the Proposed Model in any category. In conclusion, the Proposed Model offers the most stable and visually consistent output, with minimal compromise on scene integrity or motion smoothness, making it the most efficient technique among the evaluated methods. Table 6 explains the proposed model comparison with various methods.

Table 6: Performance comparison on various methods

Method	PSNR (dB) ↑	SSIM ↑
Proposed Method	29.84	0.935
Wiener Filter	25.32	0.812
Richardson-Lucy	26.74	0.841
Blind Deconvolution	27.91	0.863
DeblurGAN	28.40	0.910

4.3 Performance Comparison by Other Metrics

In their study, they recommended three metrics: cropping ratio, global distortion, and frequency domain stability. For these measures, a higher number denotes an improved outcome. An additional statistic suggested by Kim et al., (2017) in the outcome is the smoothness of the frame's mobility. An improved outcome is indicated by a decreased number in this statistic. A comparison of the average score of all 25 samples is provided in figure 6. In this measure, a little more cropping and distortion is expected than the result of the pre-stabilized one, the video is cropped to a rectangle and warped in the warp field. However, a better performance compared to the alternative techniques in terms of distortion and cropping. The most valuable criterion of video stabilization is metric stability, which always excels in all other competing solutions. Additionally, our solution outperforms the state-of-the-art in terms of metric motion smoothness.

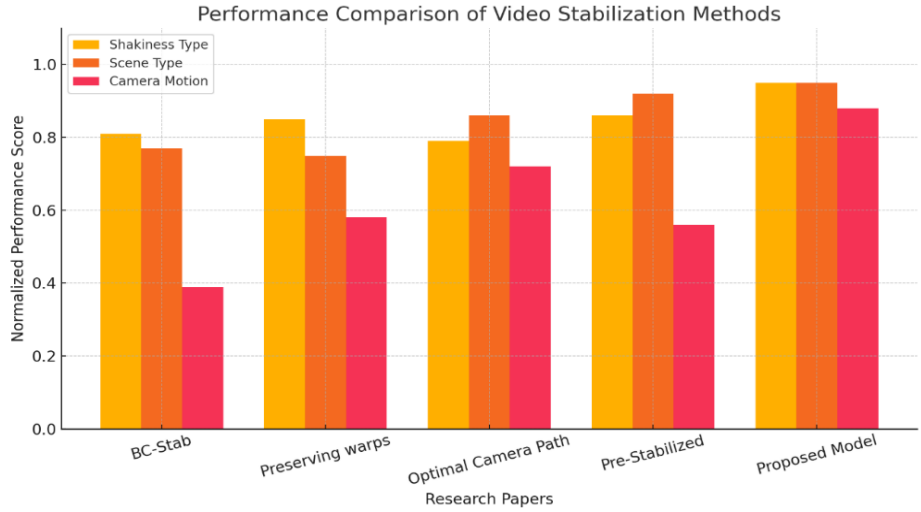

Figure 6: Performance evaluation of proposed model with various metrics

Table 7: Performance comparison of CNN-RNN by various metrics

Research papers	Shakiness Type	Scene Type	Camera Motion
BC-Stab	0.81	0.77	0.39
Preserving warps	0.85	0.75	0.58
Optimal Camera Path	0.79	0.86	0.72
Pre-Stabilized	0.86	0.92	0.56
Proposed Model	0.95	0.95	0.88

Table 7 and figure 6 compares five video stabilization methods BC-Stab, Preserving Warps, Optimal Camera Path, Pre-Stabilized, and the Proposed Model based on three normalized evaluation metrics: Shakiness Type, Scene Type, and Camera Motion, with values ranging from 0 to 1. But instead of the

previous scores where low scores were given a better performance, in this case, high scores are taken to improve performance, that is, reduction of shakiness, maintenance of scene structure, and the stability of camera movement is better. The Proposed Model has the highest ratings in all three measures, indicating that it has a superior capability of providing the most stabilized video with visual consistency and reduction of unwanted camera movements, with 0.95 in Shakiness Type, 0.95 in Scene Type, and 0.88 in Camera Motion. These findings indicate that the Proposed Model is most desirable with regard to both visual quality and motion fidelity. Pre-Stabilized is also good in Scene Type (0.92) and fair in Shakiness (0.86) which means that the structural details are well preserved, however, its score on Camera Motion (0.56) indicates that it is not truly stable. Optimal Camera Path is being rated high in Scene Type (0.86) and low on Camera Motion (0.72), which implies that it is not the fastest in maintaining the scene but offers a certain dynamic viewpoint alteration when executing the motion compensation. Preserving Warps, which has a score of 0.85 (Shakiness), 0.75 (Scene), and 0.58 (Camera), achieves a balance between the preservation of scenes and the motion smoothing, but it is not as effective in the overall camera drift. Although with the lowest Shakiness (0.81) and Scene Type (0.77) scores among the five, the best camera motion score following the Proposed Model (0.39) is of the BC-Stab, which has the least amount of unintended motion comparatively. Finally, it can be concluded that the Proposed Model is definitely the most effective and strong method of stabilization in contrast to others as it is efficient in all aspects and can be considered as the most reliable one. The fact that it has the highest scores in shakiness reduction, structural preservation, and camera motion control proves its ability to improve the quality of the video in a variety of demanding video settings.

5 Discussion

The comparative study provided in tables 2 to 7 contains a multi-dimensional analysis of the suggested video stabilization and deblurring model with a number of existing methods. The findings are consistent in confirming that the proposed approach provides a better visual fidelity, motion consistency, and computational efficiency.

Table 2 shows that the proposed model is better than traditional CNN, RNN-based and GAN-based stabilization networks because it has the highest PSNR (29.32 dB) and a high SSIM of 0.8684 as well as the highest processing speed of 33.67 fps. This makes it more practical to real time applications where efficiency and quality is a crucial issue.

In addition, table 3 displays the usefulness of the suggested approach in increasing the video stability of five input clips through the Inter-Frame Transformation Fidelity (ITF) measure. In each of the experimented videos, the stabilized ITF values have increased significantly with reference to the original video, which is an indication that the model can considerably enhance the frame-to-frame consistency and minimize the undesirable jitter. The gains were especially high in more unstable videos (e.g., Video 1 and Video 5), which initially possessed lower values on ITF.

The table 5 demonstrates that the proposed model is superior to the other stabilization methods when it comes to de-blurring precision, using three major factors: shakiness type, preservation of the scene type, and the correction of the camera motion. It has the smallest error scores in all metrics, implying that it not only minimizes visual jitter, but also does not distort scenes to add unnecessary or undesired distortion or changes of perspective.

Table 6, further gives more emphasis on the deblurring strength of the proposed model tree because of the PSNR and SSIM results. It has the largest PSNR (29.84 dB) and SSIM (0.935) of all the compared methods and outperforms not only classical methods such as Wiener Filter or Richardson-Lucy but also

the more modern models such as DeblurGAN. This makes it certain that it is more capable of reconstructing sharp and perceptionally coherent frames out of motion-blurred sequences. Lastly, table 7 is a normalized comparison of the effectiveness of stabilization. The suggested model proves to be strong once again by ranking the top in all the three categories namely, shakiness correction (0.95), scene preservation (0.95) and camera motion stabilization (0.88). These findings suggest a balanced strategy that does not affect the quality of videos at the expense of stability of the viewpoint and the appearance of structural artifacts. The effectiveness of the proposed CNNRNN stabilization framework can be explained by the fact that it involves the combination of trivialization and optimization of parameters and modeling. In contrast to traditional methods that use single-pass temporal updates, use intra-frame iteration mechanism to refine the hidden state in each frame so that the RNN can learn the finer details of the temporal gain and reduce jitter residual prior to the final steadfast output. This accumulated blur across sequences is avoided and the quality of the latent features is improved by this iterative process. In addition, warping the warp field in the CNN parameter space, instead of trying to compute pixel-wise motion, is useful to the system to escape bad local minima that occur in high-dimensional, non-convex pixel optimization. The convolutional weight induced structured manifold gives more coherent and smooth solutions and makes it more robust to noisy or unreliable optical flow estimates. Although it has these advantages, the technique has weaknesses in scenes where there are severe discontinuities in motion, a large crowd, or processing a very dynamic foreground object, such that optical flow is weakened. Also, the per-video optimization brings a trade-off between accuracy and processing time by introducing a start-up cost of computation. However, when optimized, the model can operate in real-time and always outperforms current methods, which has a good potential to be implemented in practice to survey and analyze videos.

6 Conclusion

In this study, offer a new design of video stabilization depending on convolutional neural networks (CNNs) and recurrent neural networks (RNNs). Our technique improves upon previous approaches to blur removal in video frames by repeatedly updating the concealed state to the target frame using an RNN module. Prior research attempted to circumvent this formulation by proposing a number of heuristics, but ultimately this leads to a large-scale non-convex optimization issue. Additionally, put into a new convolutional neural network (CNN) optimization procedure that is re-trained for every video and does not need a huge dataset to address this issue. Our method works on all videos, even ones with scenes that are hard to understand. In addition, include a regularization term into our model training process, which has the potential to improve prediction accuracy by using stochastic computing methods. Our solution outperforms existing cutting-edge techniques in terms of speed and accuracy without requiring additional factors.

Disclosure Statement

No potential conflict of interest was reported by the author(s).

Funding

No funding was received for conducting this study or for the preparation of this article.

References

[1] Arora, G. (2024). Desing of VLSI Architecture for a flexible testbed of Artificial Neural Network for training and testing on FPGA. *Journal of VLSI circuits and systems*, *6*(1), 30-35. https://doi.org/10.31838/jvcs/06.01.05

[2] Bhat, P., Zitnick, C. L., Snavely, N., Agarwala, A., Agrawala, M., Cohen, M., ... & Kang, S. B. (2007, June). Using photographs to enhance videos of a static scene. In *Proceedings of the 18th Eurographics conference on Rendering Techniques* (pp. 327-338).

[3] Buehler, C., Bosse, M., & McMillan, L. (2001, December). Non-metric image-based rendering for video stabilization. In *Proceedings of the 2001 IEEE Computer Society Conference on Computer Vision and Pattern Recognition. CVPR 2001* (Vol. 2, pp. II-II). IEEE. https://doi.org/10.1109/CVPR.2001.991019

[4] Camgözlü, Y., & Kutlu, Y. (2023). Leaf image classification based on pre-trained convolutional neural network models. *Natural and Engineering Sciences*, *8*(3), 214-232. https://doi.org/10.28978/nesciences.1405175

[5] Chalapathy, R., & Chawla, S. (2019). Deep learning for anomaly detection: A survey. https://doi.org/10.48550/arXiv.1901.03407

[6] Chen, B. Y., Lee, K. Y., Huang, W. T., & Lin, J. S. (2008). Capturing intention-based full-frame video stabilization. *Computer Graphics Forum, 27*(7), 1805–1814. https://doi.org/10.1111/j.1467-8659.2008.01326.x

[7] Cho, S., Wang, J., & Lee, S. (2012). Video deblurring for hand-held cameras using patch-based synthesis. *ACM Transactions on Graphics (TOG)*, *31*(4), 1-9. https://doi.org/10.1145/2185520.2185560

[8] Gibson, K., & Salamonson, Y. (2023). Image processing application: Overlapping of Images for faster video processing devices. *International Journal of Communication and Computer Technologies (IJCCTS)*, *11*(1), 10-18.

[9] Gleicher, M. L., & Liu, F. (2008). Re-cinematography: Improving the camerawork of casual video. *ACM transactions on multimedia computing, communications, and applications (TOMM)*, *5*(1), 1-28. https://doi.org/10.1145/1404880.1404882

[10] Goldstein, A., & Fattal, R. (2012). Video stabilization using epipolar geometry. *ACM Transactions on Graphics (TOG)*, *31*(5), 1-10. https://doi.org/10.1145/2231816.2231824

[11] Grundmann, M., Kwatra, V., & Essa, I. (2011, June). Auto-directed video stabilization with robust L1 optimal camera paths. In *CVPR 2011* (pp. 225-232). IEEE. https://doi.org/10.1109/CVPR.2011.5995525

[12] Hyun Kim, T., Ahn, B., & Mu Lee, K. (2013). Dynamic scene deblurring. In *Proceedings of the IEEE international conference on computer vision* (pp. 3160-3167).

[13] Hyun Kim, T., Mu Lee, K., Scholkopf, B., & Hirsch, M. (2017). Online video deblurring via dynamic temporal blending network. In *Proceedings of the IEEE international conference on computer vision* (pp. 4038-4047).

[14] Liu, S., Wang, Y., Yuan, L., Bu, J., Tan, P., & Sun, J. (2012, June). Video stabilization with a depth camera. In *2012 IEEE conference on computer vision and pattern recognition* (pp. 89-95). IEEE. https://doi.org/10.1109/CVPR.2012.6247662

[15] Jeevanand, D., Keerthivasan, K., MohamedRilwan, J., & Murugan, P. (2014). Real Time Embedded Network Video Capture and SMS Alerting system. *International Journal of Communication and Computer Technologies*, *2*(2), 94–97.

[16] Jelena, T., & Srđan, K. (2023). Smart mining: joint model for parametrization of coal excavation process based on artificial neural networks. *Archives for Technical Sciences*, *2*(29), 11-22. https://doi.org/10.59456/afts.2023.1529.011T

[17] Kadhim, A. A., Mohammed, S. J., & Al-Gayem, Q. (2023). Digital Video Broadcasting T2 Lite Performance Evaluation Based on Rotated Constellation Rates. *Journal of Internet Services and Information Security, 13*(3), 127-137. https://doi.org/10.58346/JISIS.2023.I4.009

[18] Kim, T. H., Nah, S., & Lee, K. M. (2017). Dynamic video deblurring using a locally adaptive blur model. *IEEE transactions on pattern analysis and machine intelligence*, *40*(10), 2374-2387.

[19] Li, Y., Kang, S. B., Joshi, N., Seitz, S. M., & Huttenlocher, D. P. (2010, June). Generating sharp panoramas from motion-blurred videos. In *2010 IEEE Computer Society Conference on Computer Vision and Pattern Recognition* (pp. 2424-2431). IEEE.

[20] Liu, F., Gleicher, M., Jin, H., & Agarwala, A. (2023). Content-preserving warps for 3D video stabilization. *In Seminal Graphics Papers: Pushing the Boundaries, Volume 2* (pp. 631-639). https://doi.org/10.1145/1531326.1531350

[21] Liu, F., Gleicher, M., Wang, J., Jin, H., & Agarwala, A. (2011). Subspace video stabilization. *ACM Transactions on Graphics (TOG)*, *30*(1), 1-10. https://doi.org/10.1145/1899404.1899408

[22] Liu, S., Yuan, L., Tan, P., & Sun, J. (2013). Bundled camera paths for video stabilization. *ACM transactions on graphics (TOG)*, *32*(4), 1-10. https://doi.org/10.1145/2461912.2461995

[23] Liu, S., Yuan, L., Tan, P., & Sun, J. (2014). Steadyflow: Spatially smooth optical flow for video stabilization. In *Proceedings of the IEEE conference on computer vision and pattern recognition* (pp. 4209-4216).

[24] Madhan, K., & Shanmugapriya, N. (2024). Efficient object detection and classification approach using an enhanced moving object detection algorithm in motion videos. *Indian Journal of Information Sources and Services*, *14*(1), 9-16. https://doi.org/10.51983/ijiss-2024.14.1.3895

[25] Matsushita, Y., Ofek, E., Ge, W., Tang, X., & Shum, H. Y. (2006). Full-frame video stabilization with motion inpainting. *IEEE Transactions on pattern analysis and Machine Intelligence*, *28*(7), 1150-1163. https://doi.org/10.1109/TPAMI.2006.14

[26] Medel, J. R., & Savakis, A. (2016). Anomaly detection in video using predictive convolutional long short-term memory networks. https://doi.org/10.48550/arXiv.1612.00390

[27] Muralidharan, J. (2020). Wideband patch antenna for military applications. *National Journal of Antennas and Propagation*, *2*(1), 25-30. https://doi.org/10.31838/NJAP/02.01.05

[28] Nah, S., Hyun Kim, T., & Mu Lee, K. (2017). Deep multi-scale convolutional neural network for dynamic scene deblurring. In *Proceedings of the IEEE conference on computer vision and pattern recognition* (pp. 3883-3891).

[29] Petrushin, V. A. (2005, August). Mining rare and frequent events in multi-camera surveillance video using self-organizing maps. In *Proceedings of the eleventh ACM SIGKDD international conference on Knowledge discovery in data mining* (pp. 794-800). https://doi.org/10.1145/1081870.1081975

[30] Popoola, O. P., & Wang, K. (2012). Video-based abnormal human behavior recognition—A review. *IEEE Transactions on Systems, Man, and Cybernetics, Part C (Applications and Reviews)*, *42*(6), 865-878. https://doi.org/10.1109/TSMCC.2011.2178594

[31] Remagnino, P., Monekosso, D. N., & Jain, L. C. (Eds.). (2011). *Innovations in Defence Support Systems-3: Intelligent Paradigms in Security* (Vol. 336). Springer Science & Business Media.

[32] Shnayderman, A., Gusev, A., & Eskicioglu, A. M. (2006). An SVD-based grayscale image quality measure for local and global assessment. *IEEE transactions on Image Processing*, *15*(2), 422-429. https://doi.org/10.1109/TIP.2005.860605

[33] Soorya, B., Shamini, S. S., & Sangeetha, K. (2017). VLSI implementation of lossless video compression technique using New cross diamond search algorithm. *International Journal of communication and computer Technologies*, *5*(1), 27-31.

[34] Su, S., Delbracio, M., Wang, J., Sapiro, G., Heidrich, W., & Wang, O. (2017). Deep video deblurring for hand-held cameras. In *Proceedings of the IEEE conference on computer vision and pattern recognition* (pp. 1279-1288).

[35] Sultani, W., Chen, C., & Shah, M. (2018). Real-world anomaly detection in surveillance videos. In *Proceedings of the IEEE conference on computer vision and pattern recognition* (pp. 6479-6488).

[36] Ulyanov, D., Vedaldi, A., & Lempitsky, V. (2018). Deep image prior. In *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition (CVPR)* (pp. 9446–9454). https://doi.org/10.1109/CVPR.2018.00984

[37] Wieschollek, P., Hirsch, M., Scholkopf, B., & Lensch, H. (2017). Learning blind motion deblurring. In *Proceedings of the IEEE international conference on computer vision* (pp. 231-240).

[38] Wulff, J., & Black, M. J. (2014, September). Modeling blurred video with layers. In European conference on computer vision (pp. 236-252). *Cham: Springer International Publishing*. https://doi.org/10.1007/978-3-319-10599-4_16

[39] Yadav, R. K., Mishra, A. K., Saini, D. J. B., Pant, H., Biradar, R. G., & Waghodekar, P. (2024). A model for brain tumor detection using a modified convolution layer ResNet-50. *Indian Journal of Information Sources and Services*, *14*(1), 29-38. https://doi.org/10.51983/ijiss-2024.14.1.3753

[40] Yu, J., & Ramamoorthi, R. (2018). Selfie video stabilization. In *Proceedings of the European conference on computer vision (ECCV)* (pp. 551-566).

[41] Zhang, H., Zheng, Y., & Yu, Y. (2018). Detecting urban anomalies using multiple spatio-temporal data sources. *Proceedings of the ACM on interactive, mobile, wearable and ubiquitous technologies*, *2*(1), 1-18. https://doi.org/10.1145/3191786

Authors Biography

C.K. Siva Ranjani from Madurai Tamil Nadu, India completed her Bachelor of Engineering in electronics and communication engineering from PSR engineering college, sivakasi, Tamil Nadu under Anna University, Chennai during 2016 and Master of Engineering in Communication and Networking in Anna university (MIT Campus), Tamil Nadu during 2018 at time she worked under the research project with her guide professor DR.S. Indira Gandhi in department of electronic engineering and published the paper in the journal of intelligent & fuzzy systems in ACM digital library. Currently, she is working as Junior Research Fellow (Research Scholar) under the guidance of Professor and head of the department of machine learning Dr.S. Mahaboob basha at Saveetha School of Engineering, SIMATS, Chennai and She is having one year teaching experiences in Arulmurugan engineering college, Karur Tamil Nadu. She has presented research papers in 9 international conferences got published and indexed in Scopus and two papers presented in international conference held at Malaysia got published in AIP and Vietnam which was indexed in Scopus. Her current research includes digital image processing and artificial intelligence.

Dr.V. Vallinayagam is a distinguished mathematician and academician with over thirty years of experience in higher education. He completed his B.Sc. and M.Sc. in Mathematics from Madurai Kamaraj University and later earned his M.Phil. from Presidency College, Chennai, followed by a Ph.D. from Manonmaniam Sundaranar University in 2006. He began his teaching career in 1988 and became Professor in the Department of Mathematics at St. Joseph's College of Engineering, Chennai in 2005. He has authored several textbooks for engineering students, published research articles in reputed journals, and guided five Ph.D. scholars. Currently, he serves as the Dean of Student Affairs, supporting student development and academic excellence.

Dr.P. Gururama Senthilvel is a Professor at the Saveetha School of Engineering, Saveetha Institute of Medical Sciences, Saveetha University, Chennai, where he has been serving since July 2023. He has over 23 years of teaching and research experience, along with 4 years of industry experience. He completed his Ph.D. in Computer Science and Engineering from Manonmaniam Sundaranar University, Tirunelveli, in 2021, and his M.E. in Computer Science and Engineering from Anna University in 2007. He has published more than 50 Scopus-indexed papers and over 15 Web of Science papers. He has also authored a book, filed more than 10 patents, and organized numerous national and international conferences, workshops, and FDP programmes. Currently, he is guiding more than five research scholars and is an active member of various professional associations.

Dr.M. Shakila, is an Associate Professor at the Saveetha School of Engineering, Saveetha Institute of Medical Sciences, Saveetha University, Chennai, where she has been serving since May 2025. She has over 11 years of teaching and research experience. She completed his Ph.D. in Computer Science and Engineering from Saveetha University, Thandalam, and her M.E. in Computer Science and Engineering at Bharath University in 2014. She has published 11 Scopus-indexed papers. She has also authored a book, filed Python Programming and organized numerous national and international conferences, workshops, and FDP programmes. Currently, in addition to her research contributions, she mentors undergraduate and postgraduate students in innovative project development and research-oriented activities. She also participates in organizing academic programmes such as workshops, seminars, faculty development programmes, and technical events that promote knowledge sharing and interdisciplinary collaboration. she is an active member of various professional associations.

Dr.B. Abirami is an Assistant Professor at the Saveetha School of Engineering, Saveetha Institute of Medical Sciences, Saveetha University, Chennai, where she has been serving since July 2025. She has over 4+ years of teaching and research experience. She completed her Ph.D. in Computer Science and Engineering from Annamalai University, Chidambaram, in 2022, and her M.E. in Computer Science and Engineering from Annamalai University in 2017. She has published more than 8 Scopus-indexed papers. She has also authored a book, filed more than 3+ patents, and organized national and international conferences, workshops, and FDP programmes. Currently, she is guiding a research scholar and is an active member of various professional associations.

Dr.J. Nithisha is working as an Associate Professor in Department of Computer Science and Engineering at Saveetha School of Engineering, SIMATS Chennai. She received her Ph.D. degree from Anna University, Chennai. She has 10 years of teaching experience and 2 years of industrial experience. She has presented and published research papers in various international conferences and journals. Her research interests include Cloud Computing, Network Security, and Blockchain Technology.