Video Deraining for Mutual Motion by Fast Bilateral Filtering on Spatiotemporal Features

Apurba Das, Shylaja S S

Abstract: In video analytics, out of all the known challenges, (a) adverse unconstrained environmental condition resulting low illumination/ contrast and (b) mutual motion between camera assembly and object of interest are definitely the major ones. The challenges become multi-fold when the scene of interest gets impacted by both the aforementioned simultaneously. The first part of the current work has proposed a novel hierarchical scene categorization methodology based on selected spatiotemporal features. The procedure of determination of feature hierarchy and the detailed description of step by step methodology of the same has been provided along with comprehensive results. The second part of the current work proposes a novel method of processing rainy videos in different mutual motion scenarios by dynamic bilateral filtering and deep auto-encoder on time-sliced video frames. The proposed idea exploited the strength of range-domain filtering of bilateral filter, saliency extraction of deep auto-encoder and spatiotemporal model of time-slicing. Quantitative results of extensive experiments showed the effectiveness of our proposed algorithm in different degree of environmental degradation. The proposal of the scene categorization not only classifies the scenes based on mutual motion between camera assembly and object of interest, but it also triggers and influences the algorithm of deraining dynamically by offering required modulation.

Keywords: bilateral filtering, decision tree, Fuzzy C-means, motion vector, Histogram of Flow (HoF), sigmoid, time-slicing.

1. INTRODUCTION

Vision based analysis of outdoor scene has been always a subject of interest for computer vision researchers ranging from intelligent transportation system to surveillance. With the advancement of intelligent video analytics in outdoor scenes, expectation of the industry is increasing even in completely unconstrained environment in terms of non-uniform, insufficient illumination and adverse weather conditions [1], [2] like fog, rain, snow etc. The current work has focused on addressing negative impact of rain streaks on video quality and in turn the video analytics. Deraining of the videos is a challenging task because of rain streak’s random patterns in frames and dynamic modulation of the said pattern due to possible motion of camera. Rain streaks or droplets in air also tend to cause blurring or hazy effect to images or videos, which impacts the accuracy of vision based object detection, tracking etc.

Rainy videos with moving cameras, static/ moving object of interest creates various challenging scenarios for rain removal or deraining filters. Approaches proposed by state-of-art researchers in video deraining worked well mostly depended on some assumptions with temporal information, but could often be distorted by motion of objects or the camera itself. Various experiments have been targeted to handle the said two possible relative motions separately. However, they rely on few assumptions like accurate extraction of foreground, etc. Hence, the said approach may not be as effective in restoring videos to a sufficiently good extent always by filtering the complex noise induced by rain streaks in dynamic video frame sequences addressing all possible relative motions between the acquisition system and subject of interest.

Therefore, considering the challenges involved in the problem of rain streak removal effectively in both static and dynamic conditions, in this paper, we propose a novel approach to achieve effective video restoration and enhancement by a spatiotemporal composition of frame sequences named as time-sliced images [3], [4]. In this paper, we addressed various mutual motion problems in video between camera and object of interest by two independent approaches with distinguishable accuracy-performance trade-off namely: (a) Bilateral filtering based approach and (b) Deep convolutional auto-encoder based approach on time-sliced images.

Our target revolves around deraining videos captured under the following circumstances:

• Static camera, static object (SCSO or CSM).
• Static camera, moving object (SCMO or CSMO).
• Moving camera, static object (MCSO or CMSO), and
• Moving camera, moving object (MCMO or CMOM).

Classifying different scenarios and detecting objects in these scenarios are important to be addressed in order to design dynamic filter for identified aforementioned mutual motion scenarios.

Ng et al. [5] classified complete motion stream by deep learning. Mishra et al. [6] proposed motion based classification where classification of vehicles is done using feature-based algorithm. Zhang et al. [7] and Kim et al. [8] analyzed object classification of diverse camera viewing angles. As Mitra et al. [9] used optical flow for expression clustering. Agarwal et al. [10], Starkov et al. [11], and Thakoor et al. [12] used the same to detect moving objects with static camera in any background or terrain.
Video Deraining for Mutual Motion by Fast Bilateral Filtering on Spatiotemporal Features

The aforementioned works, addressed the motion compensation assuming constant mutual motion between camera and object. They have certain limitations, failing to handle multiple scenarios. In this paper, we proposed an algorithm to automatically classify the type of motion available in the scene. First, we have generated the sparse optical flow of the input scene and then computed various properties based on those good points provided by the optical flow. Among all the properties, experiments have proved that magnitude and phase direction are the best features to accurately classify the type of motion prevalent in the scene. Finally, Fuzzy C-Means clustering algorithm has been adopted to identify the motion present in the video.

Our proposed algorithm of deraining has exploited the strength of (a) time-slicing [3] by spatiotemporal combined representation and (b) bilateral filtering [13][14][15] where psycho-visual prominence along with spatial variation, i.e., range-domain simultaneous processing has been taken care.

An earlier research carried out by Bossu et al. [16] proposed photometry based selection of candidate rainy pixels. Tripathi et al. [17] proposed a probabilistic model which is robust to variations in intensity of rainfall. In general, most of these approaches perform well with static backgrounds. However, they either have structural and orientation constraints of rainfall for which they may not perform effectively for moving objects with overlapping rain pixels and hence may not be suitable for dynamic scenes with low, medium and high-density rain. With the aim of addressing the above-mentioned problems, Chen et al. [18] proposed motion-based segmentation. In a recent research experiment, Ren et al. [19] created a framework for matrix decomposition to model detection and removal of rain streaks. Recently, Chen et al. [20] proposed the alignment of scene content at the super pixel level comparable to other state-of-art like [21]. It was inferred in their research that video based algorithms were more effective in rain removal and established temporal correspondence well, however their performance was drastically hindered under rapid motion of the camera. Therefore, with the motive of overcoming the aforementioned shortcomings, our research is aimed at creating a robust model for restoring videos after rain removal, for effective deraining of frames in all cases of motion of both camera or background.

This paper has been organized as follows. In Section II, the fundamental idea of using histogram of flow (HoF) as a scene designator has been presented through proposed hierarchical clustering. The proposed methodology of deraining has been described in Section III for various aforementioned mutual motion scenarios. The results obtained from our experiments have been studied and elaborated extensively in Section IV. Finally, Section V concludes our work and provides a brief insight about future direction which can be explored to extend this research.

II. SCENE DESIGNATION THROUGH HIERARCHICAL CLUSTERING

The aforementioned categorization of scenes from input videos has been addressed by focusing on both temporal and spatial features. Sample flow vectors for each of the categories have been shown in Fig. 1. Looking at the pattern from the representative videos for each of the four aforementioned classes, it’s evident that single-shot categorization is challenging. It’s also observed that the magnitude of flow vectors plays an important role in primary segregation but the same feature is not sufficient to segregate all 4 classes. In real-life it is most likely to get noisy or unwanted flow vectors which might disturb the defined features, e.g., in SCMO scenario recorded in Fig. 1, camera is static but still there is a slight motion in camera too. Due to which a jiggling motion is found which creates the optical flow for that motion the same behavior is observed even for MCMO. The following features have been observed to be useful in the said categorization in different levels but not in single step:

- Variance of flow magnitude (VarOfMotionFlow)
- Mean magnitude change in frames and vectors (MeanD)
- Weighted HoF (WHoF)

Keeping the said considerations in mind, a decision tree [23] has been employed to identify most relevant and suitable feature for hierarchical categorization of scenes.

A. Decision Tree Based Feature Hierarchy and Selection

In order to derive the most suitable features from the feature bank related to flow vectors and frames of the input videos and hierarchical relationship among them, Decision tree [23] has been employed. The decision tree depicted in Fig. 2 has shown, the variance of flow vector can classify the static object videos from moving object videos.

We have used the selected features through decision tree even though the chronology is not followed, in order to achieve child node quickly with depth = 3 and inequality measure as Gini score. In the Fig. 2, X₀, X₁, X₂ represent variance of motion flows, (VarOfMotionFlow), Weighted HoF (WHoF) and Mean of magnitude change across frames and across vectors (MeanD), respectively.

In first layer, X₂ could classify MCMO class from other classes. Next, X₁ was able to classify the MCSO from other classes. The remaining classes, SCSO and SCMO could be classified by X₀.

B. Flow Vector and HoF

Optical flow [24] represents a design of probable movement of objects generated by relative motion between object and observer in the target scene.
Fig. 1. Flow vector for the 4 mutual motion categories from dataset [22]: (a) SCSO frame, (b) SCSO flow vector clustered, (c) SCMO frame, (d) SCMO flow vector clustered, (e) MCSO frame, (f) MCSO flow vector clustered, (g) MCMO frame, and (h) MCMO flow vector clustered.

Fig. 2. Decision tree for feature selection and hierarchy

The optical flow gives us information about the movement of object of interest (flow vectors) among series of frames produced by the motion of camera and also provides us estimation of object position in corresponding frames which is robust enough to predict the state of the object, whether the object is static or moving. Sparse optical flow is extracted here on each of the frames where points are randomly generated in initial frame on detected corners and then those points are tracked in the corresponding frames. In our experiment, we have performed optical flow by using the iterative Lucas-Kanade method [24] with pyramids on the 100 features points detected on the frame captured from the scene with the help of Harris Detector. These selected points and both the frames (current and upcoming next) of the scene are sent to the optical flow. The optical flow will provide feature points present in the next frame. Based on these points for two consecutive frames, various features are derived which plays a vital role in the classification of the video.

The different properties computed using these feature points are as follows:
1) Position (x, y) of the feature point in the current frame.
2) Position (x, y) of the new feature point present in the new frame that tracks to the corresponding feature point present in the previous frame.
3) Euclidean distance, d between the two same feature points present in the current and new frames.
4) Angle formed between the two same feature points.
present in the current and new frames.
5) Moving average point, \((M_x, M_y)\) has been calculated in every frame of the scene for each of the feature points following the Eq. 1.
6) Euclidean distance between the two-moving average point with respect to current point resent in the current and new frames.
7) Angle formed between the two-moving average point of same feature points present in the current and new frames.
8) Total distance traveled by each of the feature points from first frame to the last frame in the scene.

For the first frame of a scene, \(M_x\) and \(M_y\) is equal to the position \((x, y)\) of the feature point in the first frame. But for the second frame onward, \(M_x\) and \(M_y\) is computed as depicted in Eq. 1a and 1b. Here, \((P_x, P_y)\) and \((M_x, M_y)\) are represented as the feature point of the current frame and the moving average point of that feature point in the next frame, respectively.

\[
M_x = 0.8 \times M_x + 0.2 \times P_x \quad (1a)
\]

\[
M_y = 0.8 \times M_y + 0.2 \times P_y \quad (1b)
\]

It’s obvious that the behavior of the optical flow would vary depending on the type of mutual motion available in the scene. Ideally in case of static camera and static object scenario, the tracking feature points should have no difference across all the frames as all the points will be always static. On the other hand, for video stream captured from an ego vehicle either stationary or in motion, there is a non-linearity observed for the optical flow magnitudes: small near vanishing points and large near the camera as depicted in [25]. The current paper has addressed the aforementioned feature by employing both position and flow vector simultaneously for all calculations. This idea of vanishing point estimation and degree of convergence of optical flow vectors [25] has been instrumental of deriving the depth of object of interest. Still, unexpected flow vectors have been observed depicting jiggling motion or noise in the scene. The same has been removed by employing bilateral filter [26] on the computer flow vector domain and not on the original video frames. The noisy flow vectors and their corresponding denoised flows have been depicted in Fig. 3. Finally, histogram of flow (HoF) has been calculated following the method depicted by Das et al. [25] for future use of the same as a feature for hierarchical categorization.

### A. Fuzzy C-means (FCM) Clustering

Overlapping of classes occur due to the uncertainty inherently present in any natural data starting from image/video to textual/speech data in most of the practical classification problems. In the present problem, scene categories are also overlapping in nature. Due to the presence of inherent uncertainty in the class boundary, we have applied the Fuzzy C-means clustering algorithm [9] here. In the Fuzzy C-means clustering it is assumed that each sample \(x_i\) has some graded or, fuzzy memberships in a cluster. These memberships are equivalent to the probabilities \(P(\omega_j|x_i, \theta)\), where \(\theta\)theta is the parameter vector for the membership function. The Fuzzy-C means clustering (FCM) algorithm seeks a minimum of a heuristic global cost function \(J_{fuz}\) as depicted in Eq. 1.

\[
J_{fuz} = \sum_{i=1}^{c} \sum_{j=1}^{n} P(\omega_j|x_i, \theta) \left( x_i - \mu_j \right) \quad (2)
\]

where, \(b\) is a free parameter chosen to adjust the blending of different clusters. The algorithm returns \(C\) numbers of minimally overlapped clusters depending on equivalent membership criteria.

![Fig. 3. Denoising Flow vectors in motion plane: Noisy and denoised flow planes for SCSO, MCSO, MCMO and SCMO categories](image)

### B. Scene categorization Algorithm

Based on the previous subsections II-A, II-B, and II-C the detailed methodology has been formulated as follows:
1) Randomly select 100 frames from each of the 100 sample videos of the 4 mutual-motion categories.
2) Restrict computing optical flows around 100 Harris corners for each of the initial frames of the 100 videos for each class.
3) Hence, total 10,000 vectors will be generated across frames tracking the aforementioned initial corners.
4) For each of those 10,000 points, direction, magnitude, Euclidean distance across frames etc. have been calculated.
5) The flow vectors have been denoised by Bilateral filter [26]. For dimension reduction, variance of each pre-clustered members has been considered as preliminary features. Further minority clusters having <5% carnality elements have been eliminated.
6) Variance of motion flows for all frames and all videos have been used as 1st derived cluster. This feature, VarOfMotionFlow is to be ideally zero for SCSO case.
as there is no absolute or mutual motion expected. On the other hand, for MCSO also, only mutual motion is expected which makes the variance again close to zero. For any other scene designation like MCMO or SCMO, the variance is expected to be non-zero.

7) Hence, this feature, VarOfMotionFlow has been used for segregating all scenes into two groups one having SCSO+MCSC and other containing SCMO+MCMO by FCM, as depicted in Fig. 4. Before passing it to first layer, we filter variance values with Sigmoid function to force the inter-class distance between zero and non-zero variance high enough to classify in Layer 1.

8) In the next layer (Layer 2) of clustering, mean value of the total distance traversed by moving average points with respect to current point for each flow vectors (MeanD) have been treated as feature. These mean values have been passed through sigmoid function to make the inter-class distance large enough before feeding to Layer 2 of FCM.

9) Output of group 1 (having zero VarOfMotionFlow) of Layer 2 of FCM would be 2 clusters having SCSC with low MeanD and MCMO with high MeanD.

10) Next, the group 2 of Layer 1 (i.e., Layer 2) having non-zero VarOfMotionFlow is clustered based on summation of moving average point for each of the feature points. This weighted histogram of flow vectors has been treated as feature for categorizing SCMO and MCMO though adaptive thresholding of the WHoF from its modalities (Fig. 4)

**Fig. 4. Hierarchical Clustering Layers for Scene Categorization**

### III. PROPOSED ALGORITHM FOR VIDEO DERRAINING

For videos captured through stationary camera, the motion property of rain is perfectly defined as vertically down along one narrow column strip of video across frame. For videos captured through cameras in motion, the motion property of rain is relatively complex but still possible to model. The said motion property has been honored and modeled through classical and adaptive time slicing of videos as discussed in subsection III-A. On the other hand, the appearance of rain is like a noise in video. This noise removal has been addressed by two parallel and independent ways: (a) Range-domain (i.e., Bilateral) filtering to remove rain ensuring image information sustenance, (b) Deep Auto-encoder to be trained with saliency of image and derain video considering rain streaks as non-salient. User may decide out of two proposed methods, based on the performance budget and accuracy target for cleaning videos in adverse weather condition. The detailed methodology proposed in this work has been depicted in Fig. 6 and elaborated in further subsections.

#### A. Time Sliced (TS) Image Synthesis from Video Frames

Time slicing refers to a method of combining multiple frames corresponding to a scene, captured at different times, to form a single composite image representing rows as single slice of frame across time [3], [4]. As motion of rain streaks is almost vertically down for static camera scenarios, we have decided to slice vertical columns from subsequent frames of a video to synthesize TS-image (time-sliced image) for each slice across time as depicted in Eq. 3, 4 and Fig. 5.

\[ V_{lep} = I_{frm}(x, y, t), 0 \leq x \leq 639 \]

\[ 0 \leq y \leq 479 \]

\[ 0 \leq t \leq 111 \]  

(3)

\[ I_{TS}(x, y) = I_{frm}(x = x_i, y, t = 0:111); x_i \in (0, 639) \]

(4)

**Fig. 5. Synthesizing TS-image for deraining: Stacked vertically sliced column \( x = x_i \) from all video frames representing horizontal direction as time and vertical direction as spatial pixels**

As depicted in Eq. 3, the input video, \( V_{lep} \) has been represented as series of frames, \( I_{frm}(x, y, t) \) of size 640 × 480 × 112 interpreting 640 columns: x, 480 rows: y, each having 3 color separations (R, G, B) and 112 number of frames: t. One time-sliced (TS) image is created from each identified column \( x = x_i \) where \( x_i \) is a constant. Likewise, for 640 columns, total 640 TS-images, \( I_{TS}(x, y) \) have been created following Eq. 4, with different values of \( i = 0:639 \). The TS-image, \( I_{TS}(x, y) \) has been expressed in terms of \( I_{frm}(x, y, t) \) in Eq. 4 where all the time values for one specific column has been stacked as columns in the image considering all row pixels for each column. Each arbitrary time-sliced image addresses one arbitrary column throughout 112 number of frames. Parallel threads are spawn to process the other arbitrary columns simultaneously. Finally, spatiotemporal interpolation is employed to address other unattended rows.
Since each time sliced image reflects the variation of a specific column over time as a smooth transition frame, applying edge preserving denoising algorithms on this image preserves the structure of the ground truth frame upon reconstruction by inverse mapping of derained columns back to the original frames. Hence using this approach, the loss of structural pixel information is considerably minimized as compared to denoising techniques directly on frames.

1) Adaptive TS images for mutual motion scenarios

The method of time-slicing is relatively simpler and straight forward when (a) the object of interest can be perceived as static and/or (b) the object of interest is having a defined shape or shape modulation as depicted by Das et al. [3]. But in the current work, where are dealing with all possible mutual motion scenarios in a scene, the general time slides image might not be really useful as far as the rain streak removal from video is concerned. For static camera scenarios (i.e., C_{O}S and C_{O}M), we still follow the method described in Subsection III-A. But specifically, for the camera in motion scenarios (i.e., C_{O}M and C_{O}OM), further level of adaptation has been employed from scene categorization module to achieve improved deraining.

Primarily, the key frame has been extracted from the complete rainy videos which represents the master frame with respect to which the consecutive frames get the significant motion maps. The key frames are decided by squared difference between gray converted histograms (\(H_{\text{one}}(\text{intensity}=i)\)) of periodic image frames as depicted in equation 5 and 6. The default period is defined as 1 second and user has provision to change the interval (T) from dynamic configuration file used during execution. The frame at time=(t+T) \(\theta\) is key frame if Eq. 5 satisfies i.e., the squared difference between periodic histogram is higher than the pre-defined threshold, \(Th\). Any video is being grouped in terms of key frame and following non-key frames like this and the further groups are similarly formed by following the same aforementioned rule.

\[
\text{Keyframe} = \sum_{i=0}^{255} |H_{i}(i) - H_{i+T}(i)|^2 > Th \tag{5}
\]

\[
t = t + T \tag{6}
\]

Next, the corners and Eigen feature representing pixels are being identified from the key frame. As described in the Section II, optical flow vectors have been calculated on the said salient pixels as those are potential pixels through which best tracking should be performed. That's why the said pixels are designated as pixels having 'good features to track'. Next, for each of those salient pixels, correspondence map has been established across following non-key frames. Further, the low pass filtered envelope of the temporal shift has been normalized through inverse translation of the vertical slices (i.e., columns) of the video. Finally, the vanilla time-slicing has been performed on the temporally normalized video. This second level adaptation in turn improves the accuracy of deraining algorithm as depicted in Table 1. The method of adaptive time slicing for mutual motion scenario has been described in Algo. III.1. Here, for the sake of simplicity and without any loss of generality, one group of video frames starting with 1st key frame and multiple non-key frame has been considered. The same is applicable for the next group starting with the next key frame.

B. Range-Domain Filtering

Convolution of any image with a low-pass-filter results in smoothening and noise reduction (i.e., deraining). This is because, said filtering operation sets the value of each pixel to a weighted average of itself along with the neighboring pixel values. Though this approach is straight forward and effective in terms of time consumed, the drawback of low pass filters is the fact that filtering occurs independently at every pixel or group of pixels irrespective of rainy or clean pixels, which results in a blurred or distorted output, in which edges may not be preserved [13], in situations when pixels belonging to different objects under rapid motion in a dynamic environment, and hence are not suitable for our purpose. Therefore, to maintain perceptual quality of the image even after removing rain, we employed a nonlinear low pass filter for deraining.

In this category, we elaborate and experiment with bilateral filter [13], which is an edge-preserving, and denoising filter for images addressing both spatial domain neighborhood and psycho-visual range. It essentially segregates the low frequency component of an image which contains the most basic information from the high frequency part which possesses details about texture information. The bilateral filter is mathematically defined as follows in terms of Eq. 7 and 8:

\[
I_{\text{filt}}(x) = \frac{1}{W_p} \sum_{x \in \Omega} I(x) f_r(||I(x) - I(x)||) g_s(||x_i - x||) \tag{7}
\]

\[
W_p = \sum_{x \in \Omega} f_r(||I(x) - I(x)||) g_s(||x_i - x||) \tag{8}
\]

In the Eq. 7, \(I_{\text{filt}}\) corresponds to filtered image whereas the original input image possibly containing noise is represented by \(I\). The term \(x\) corresponds to the coordinate value of the current pixel and \(\Omega\) is the window centered at coordinate \(x\). Further, we have two kernels, namely range \(f_r\) and spatial \(g_s\) kernels for smoothing differences in intensities and coordinates, generally modeled by Gaussian.
On the frame, bilateral filtering has not been proven to be the best choice of filtering as far as the deraining is concerned. Hence, direct range-domain filtering of frames is not sufficient as far as deraining based video restoration is concerned. This re-establishes the requirement of a spatiotemporal model of adaptive time-slicing [3] (Subsection III-A) where, in one-shot, the range-domain filtering can be applied addressing motion across frames and psycho-visual variation, simultaneously.

In order to improve the restoration of frames after rain removal, such that the image information is preserved and sharp edges are not blurred out while deraining, we propose a preliminary pre-processing step termed as time slicing of videos [3] as elaborated in previous subsections.

Fig. 6. Proposed 2 path architecture of deraining methodology: (1) Bilateral filtering on adaptive TS-images: #1 – #2 – #3; (2a) Deep Convolutional Autoencoder training on adaptive (A-) TS-images: #4 and #5, (2b) Deep Autoencoder inference on A- TS images: #1 – #6 – #7 – #3

C. Inverse Mapping of derained adaptive TS-images

Utilizing the strength of the adaptive TS-image, effective deraining could be obtained as depicted in Section 4. Post filtering either by bilateral filter or by deep convolutional autoencoder, there is a need to perform the inverse mapping of the TS-images to original video frames performing the following method.

In order to inverse map the adaptive TS-images after filtering, an arbitrary column $i_{col}$ in the time sliced images is selected, which is analogous to selecting a particular frame in the video sequence. For instance, $i_{col}$ in all time sliced image number $T = j$ corresponds to column number $j$ in frame number $f = i$ in the sequence of video frames. This process is repeated until all the original frames in the given video sequence have been restored as depicted in the #1 and #3 operation of the two parallel flows of #1-#2-#3 and #1-#6-#7-#3 in Fig. 7. Extensive experimentation and evaluation of results obtained suggest that the restored video is enhanced and has lesser noise and blurring effect as compared to the case where the frames where directly convoluted. The quality evaluation results have been depicted in Fig. 7 for different rain densities.

Fig. 7. Evaluation Metrics for proposed algorithms gradually evolved from bilateral filter through TS based bilateral to deep autoencoder (DAE) for video deraining for various camera-object mutual motion and rain densities

The density of rain as depicted in Fig. 7 has been configured as follows by utilizing Adobe After Effect Software [28].

- Low Rain density (#1): 4,000 streaks per frame and streak size 1.93 mm
- Medium rain density (#2): 11,400 streaks per frame and streak size 1.99 mm
- High rain density (#3): 25,000 streaks per frame and streak size 2.17 mm

D. Deraining by Deep Denoising Autoencoders

Though the restoration of videos after rain removal is achieved considerably well by adaptive TS-based Bilateral filtering methodology, we further proceed to explore deep learning models to enhance video quality further in order to give flexibility to the end users in choice. A number of recent research experiments using deep learning architectures such as convolutional neural networks [29], denoising autoencoders [30] and generative adversarial networks [31] have been proposed for deraining videos. Inspired by the aforementioned research ventures, we propose the method of deraining based video reconstruction through TS-image based deep autoencoder architecture utilizing the strength of adaptive time-slicing and deep learning.

E. Experimental Setup

As it is close to impossible to obtain original clean ground truth scenes from rainy scenes, we have used a dataset obtained from the authors of [20] to which noise in the form of rain streaks was induced artificially by image editing techniques [28]. The dataset was split into training, cross-validation and testing components in 60: 20: 20 ratios. Total 7 ground truth videos, each having low, medium and high rain density added by [28] have been considered. This synthesizes $7 \times 3 = 21$ videos having total 2985 frames. For adaptive time-slicing, all the videos have been normalized.
to equal number of frames, 112. Hence, total number of frames is 21 × 112 = 2352. Even though the original frame size was 640×480, for processing the frames have been normalized to the size of 640x360. Next, the adaptive time-sliced images were generated for each video individually leading to total 640 numbers of 112×360 sized images. Hence, for deep convolutional autoencoder total 640×21=13440 images were made available for training, cross-validation and testing/inference according to said proportion. For testing the model performance, we have created a test dataset of 4 videos obtained from [32] by inducing artificial rain via [28], such that they are representative of the four different conditions mentioned in Section I.

![Detailed architecture of the deraining autoencoder](image)

Since the input data consists of video frames, the fully connected layers of a traditional deep autoencoder are replaced by convolutional layers, inspired by related research in using autoencoders to denoise images [33]. Using convolution layers along with max-pooling layers, the 360 × 640-dimensional input frames with three color channels are converted to 20×28 images in latent space which are 64 channels thick. This down-sampling process in which the input images are compressed into a low dimension forms the encoder. The loss function used in our model is mean squared error as it is relevant to our task. The optimizer used was Adam and binary cross-entropy as the loss function. The details about the layers used in the model and necessary hyper-parameters have been clearly portrayed in Fig. 8.

Two separate models were trained, first to learn reconstruction of frames from rainy and ground truth images and second to learn reconstruction of time sliced images from time sliced frames of the rainy and clear videos. For the former model, we proceed by performing the inference or prediction step using the saved model directly on input video frames and for the latter model, on the time sliced images previously generated. In the latter case, inverse mapping of columns to their initial frames is performed on the derained time sliced images generated by the model to reconstruct the original video (Fig. 6), derained. The loss function convergence for train and cross-validation set has been shown in Fig. 10. We used Keras with TensorFlow back-end to implement the models, which were trained on an HP Pavilion 15 notebook having an Intel Core i5-6200U processor and 8 GB RAM with NVIDIA GeForce 940M (2GB DDR3) GPU support. The model took 54.5 hours to complete 300 epochs of training using the specified hardware.

IV.EXPERIMENTAL RESULTS

A. Dataset and result of categorization of scenes based on mutual motions

The randomly sampled MPI-Sintel data set [22] contains 35 scenes, each of them are 50 frames long, for a total of 1628 frames. The rendered images have a resolution of 1024x436 pixels at 24 frames per second. The scenes span a large variety of environments, and actions. The end goal of developing this data set is to evaluate optical flow algorithms. Since static object static camera (SCSO) scenes have almost same optical flow, therefore they exclude the static scene from the data set. After performing the hierarchical categorization as depicted in Fig. 4, we have used 100 unknown samples for each of the scene categories, SCSO (#1), MCSO (#2), SCMO (#3) and MCMO (#4) to test the effectiveness of the proposed algorithm. The result has been presented in the Fig. 11.

B. Results of proposed deraining method

In order to establish the effectiveness with which our approach is capable of rain streak removal from videos, we need to evaluate the proposed methodology either manually, by human visual perception or using a machine generated evaluation metric score. Since evaluation only by visual perception does not quantify the performance of the models used numerically, we use two metrics commonly used to assess video quality, mean squared error, PSNR and SSIM scores which is the peak signal to noise ratio to compare and analyze the results obtained. The results obtained by our proposed methodology have been compared with three recent state-of-the-art approaches using deep learning architectures, namely clearing the skies [29], deep detail network [21] and Density-aware single image de-raining using a multi-stream dense network (DID-MDN) [34]. The detailed analysis of proposed deraining algorithms in terms of (a) different rain density, and (b) performance have been presented. We could see from Fig. 7 that quality of deraining has direct relationship with rain density for all the proposed methods. We also observed that deep autoencoder on time-sliced images works best among all as far as the PSNR is concerned. We observed that our proposed method of deep autoencoder or bilateral filtering on adaptive TS-images works reasonably well with respect to state-of-art as depicted in Table 1 and Fig. 9.
Table I: Evaluation metrics for proposed algorithms against state-of-art algorithms for video deraining in terms of PSNR and SSIM for various state combinations of camera and object of interest

| Deraining technique               | Camera | Objects | PSNR score | SSIM index |
|----------------------------------|--------|---------|------------|------------|
| Bilateral filtering on frames    | static | static  | 36.17      | 0.921      |
| Bilateral filtering with time slicing | static | static  | 38.9       | 0.930      |
| Convolutional DAE on frames      | static | static  | 39.1       | 0.935      |
| Convolutional DAE with time slicing | static | static  | 39.4       | 0.95       |
| Clearing the skies [29]          | static | static  | 34.868     | 0.969      |
| Deep Detail Network [21]         | static | static  | 35.97      | 0.954      |
| DID-MDN [34]                     | static | static  | 37.17      | 0.951      |
| Bilateral filtering on frames    | static | dynamic | 31.93      | 0.917      |
| Bilateral filtering with time slicing | static | dynamic | 32.4       | 0.928      |
| Convolutional DAE on frames      | static | dynamic | 32.8       | 0.861      |
| Convolutional DAE with time slicing | static | dynamic | 33.2       | 0.949      |
| Clearing the skies [29]          | static | dynamic | 29.35      | 0.945      |
| Deep Detail Network [21]         | static | dynamic | 29.75      | 0.958      |
| DID-MDN [34]                     | static | dynamic | 31.94      | 0.943      |

| Deraining technique               | Time (in sec) |
|----------------------------------|---------------|
| Bilateral filtering on frames    | 15.8963       |
| Bilateral filtering with TS      | 154.1424      |
| DAE on frames                    | 61.3026       |
| DAE with TS (time slicing)       | 147.2654      |
| Clearing the skies [29]          | 1480.8649     |
| Deep Detail Network [21]         | 1209.1054     |
| DID-MDN [34]                     | 2237.6889     |

Table II: Execution time for video deraining (384 frames of size 640×360)

V. CONCLUSION

The forth industrial revolution (Industry 4.0) is the era of IoT and cyber-physical system where vision computing is playing crucial role as far as sensing the environment and react intelligently is concerned. With the recent innovation across business verticals and research domains wherever vision computing is applicable is increasing its gamut starting from self-driving cars through medical vision computing to manufacturing and retail. Hence, sensing scene or activities or objects of interest in different environmental or illumination conditions and mutual motion scenarios between camera capture assembly and scene/object of interest is becoming very important area of research to solve business problems. In the current work, primarily the scene categorization based on mutual motion between capture assembly and object of interest is addressed which in turn helped modeling spatiotemporal scenes in effective manner. The said spatiotemporal model in turn has been shown effective in deraining video both in terms of accuracy and performance. This work, to be reported in near future, would open doors of a new area where scene conditioning would be possible in order to perform video analytics removing all unwanted motion and unexpected environmental effect through adversary networks.
Video Deraining for Mutual Motion by Fast Bilateral Filtering on Spatiotemporal Features

Fig. 9. Comparison of proposed algorithm with the state-of-art algorithms in different mutual motion between camera and object of interest: The four rows represent scenes with static camera and static background, static camera and dynamic objects, dynamic camera and static objects, dynamic camera and dynamic objects respectively in sequence from top to bottom. The four columns from left to right represent results obtained by our approach, clearing the skies [29], deep detail network [21] and DID-MDN [34]

Fig. 10. Convergence of loss function for the deep autoencoder model proposed

Fig. 11. Confusion Matrix for Scene Categorization by our proposed algorithm

REFERENCES
1. A. Das, S. Pai, T. Vinay, V. Shenoy, and S. S. Shylaja, “D2ehazing: Real-time dehazing in traffic video analytics by fast dynamic bilateral filtering,” in 2017 International Conference on. CVIP 2018, 2017.
2. J. Xie, L. Xu, and E. Chen, “Image denoising and inpainting with deep neural networks,” in Advances in neural information processing systems, 2012, pp. 341-349.
3. A. Das, S. S. Murthy, and U. Suddamalla, “Enhanced algorithm of automated ground truth generation and validation for lane detection system by M2BMT,” IEEE Transactions on Intelligent Transportation Systems, vol. 18, no. 4, pp. 996–1005, April 2017.
4. S. A. Niyogi and E. H. Adelson, “Analyzing gait with spatiotemporal surfaces,” in Proceedings of 1994 IEEE Workshop on Motion of Nonrigid and Articulated Objects. IEEE, 1994, pp. 64–69.
5. J. Y. Ng, M. J. Hausknecht, S. Vijayanarasimhan, O. Vinyals, R. Monga, and G. Toderici, “Beyond short snippets: Deep networks for video classification,” CoRR, vol. abs/1503.08909, 2015. [Online]. Available: http://arxiv.org/abs/1503.08909
6. P. K. Mishra, “A study on classification for static and moving object in video surveillance system,” 2016.
7. L. Zhang, S. Z. Li, X.-T. Yuan, and S. Xiang, “Real-time object classification in video surveillance based on appearance learning,” 2007 IEEE Conference on Computer Vision and Pattern Recognition, pp. 1–8, 2007. [16]
8. J. Kim, G. Ye, and D. Kim, “Moving object detection under free-moving camera,” in 2010 IEEE International Conference on Image Processing, Sep. 2010, pp. 4669–4672. [18]
9. S. Mitra, C. Saha, and A. Das, “Hierarchical clustering based facial expression analysis from video sequence,” in 2011 International Conference on Communication and Industrial Application, Dec 2011, pp. 1–5. [19]
10. A. Agarwal, S. Gupta, and D. K. Singh, “Review of optical flow technique for moving object detection,” in 2016 2nd International Conference on Contemporary Computing and Informatics (IC3I), Dec 2016, pp. 409–413.
11. S. Starkov and M. Lukyanchenko, “Moving object detection in video streams received from a moving camera,” in Supplementary Proceedings of the 3rd
International Conference on Analysis of Images, Social Networks and Texts (AIST2014), Yekaterinburg, Russia, October 10–12, 2014, pp. 59–63.

12. N. Thakoor and J. Gao, “Automatic object detection in video sequences with camera in motion,” 2004.

13. C. Tomasi and R. Manduchi, “Bilateral filtering for gray and color images,” in Computer Vision, 1998. Sixth International Conference on. IEEE, 1998, pp. 839–846.

14. A. Das, P. Nair, S. S. Shylaja, and K. N. Chaudhury, “A concise review of fast bilateral filtering,” in 2017 Fourth International Conference on Image Information Processing (ICIIP), Dec 2017, pp. 1–6.

15. A. Das, V. S. Shenoy, T. Vinay, and S. S. Shylaja, “A faster high-dimensional bilateral image filtering by efficient utilization of color sparseness offline,” in ASPCON 2018, Dec 2018, pp. 1–6.

16. J. Bossu, N. Hau'tie, and J.-P. Tarel, “Rain or snow detection in image sequences through use of a histogram of orientation of streaks,” International journal of computer vision, vol. 93, no. 3, pp. 348–367, 2011.

17. A. Tripathi and S. Mukhopadhyay, “Video post processing: low-latency spatiotemporal approach for detection and removal of rain,” IET image processing, vol. 6, no. 2, pp. 181–196, 2012.

18. J. Chen and L.-P. Chau, “A rain pixel recovery algorithm for videos with highly dynamic scenes.” IEEE Trans. Image Processing, vol. 23, no. 3, pp. 1097–1104, 2014.

19. W. Ren, J. Tian, Z. Han, A. Chan, and Y. Tang, “Video desnowing and dehazing based on matrix decomposition,” in Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition, 2017, pp. 4210–4219.

20. J. Chen, C.-H. Tan, J. Hou, L.-P. Chau, and H. Li, “Robust video content alignment and compensation for rain removal in a cnn framework,” arXiv preprint arXiv:1803.10433, 2018.

21. F. J. Huang, D. Zeng, Y. Huang, X. Ding, and J. Paisley, “Removing rain from single images via a deep detail network,” in IEEE Conference on Computer Vision and Pattern Recognition, 2017, pp. 1715–1723.

22. J. Wulff, D. J. Butler, G. B. Stanley, and M. J. Black, “Lessons and insights from creating a synthetic optical flow benchmark,” in Computer Vision – ECCV 2012. Workshops and Demonstrations, A. Fusiello, V. Murino, and R. Cucchiara, Eds. Berlin, Heidelberg: Springer Berlin Heidelberg, 2012, pp. 168–177.

23. ScikitLearn: dTree: https://scikit-learn.org/stable/modules/tree.HTML

24. B. D. Lucas and T. Kanade, “An iterative image registration technique with an application to stereo vision,” in Proceedings of the 7th International Joint Conference on Artificial Intelligence - Volume 2, ser. IJCAI'81. San Francisco, CA, USA: Morgan Kaufmann Publishers Inc., 1981, pp. 674–679.

25. A. Das, K. Ruppin, P. Dave, and P. Sharafdulheen, “Vehicle boundary improvement and passing vehicle detection in driver assistance by flow distribution,” in IPTA, Dec 2017, pp. 1–6.

26. A. Das, P. Nair, S. S. Shylaja, and K. N. Chaudhury, “A concise review of fast bilateral filtering,” in 2017 Fourth International Conference on Image Information Processing (ICIIP), Dec 2017, pp. 1–6.

27. S. Mitra, C. Saha, and A. Das, “Hierarchical clustering based facial expression analysis from video sequence,”, 2011 International Conference on Communication and Industrial Application, Dec 2011, pp. 1-5.

28. “Adobe after effects software: www.adobe.com/aftereffects.” [Online]. Available: www.adobe.com/AfterEffects

29. X. Fu, J. Huang, X. Ding, Y. Liao, and J. Paisley, “Clearing the skies: A deep network architecture for single-image rain removal,” IEEE Transactions on Image Processing, vol. 26, no. 6, pp. 2944–2956, 2017.

30. P. Vincent, H. Larochelle, Y. Bengio, and P.-A. Manzagol, “Extracting and composing robust features with denoising autoencoders,” in Proceedings of the 25th international conference on Machine learning. ACM, 2008, pp. 1096–1103.

31. H. Zhang, V. Sindagi, and V. M. Patel, “Image de-raining using a conditional generative adversarial network,” arXiv preprint arXiv:1701.00877, 2017.

32. “Free hd stock footage and 4k videos!: https://www.videezy.com.” [Online]. Available: https://www.videezy.com

33. L. Gondara, “Medical image denoising using convolutional denoising autoencoders,” in Data Mining Workshops (ICDMW), 2016 IEEE 16th International Conference on. IEEE, 2016, pp. 241–246.

34. H. Zhang and V. M. Patel, “Density-aware single image de-raining using a multi-stream dense network,” in CVPR, 2018.