Journal of Statistics & Management Systems
ISSN 0972-0510 (Print), ISSN 2169-0014 (Online)
Vol. 26 (2023), No. 1, pp. 225–239
DOI : 10.47974/JSMS-961

Rain detection and removal from slowly varying background videos

Pankaj Prusty
Bibhu Prasad Mohanty *
Department of Electronics and Communications Engineering
Siksha 'O' Anusandhan (Deemed to be University)
Khandagiri
Bhubaneswar
Odisha
India

Abstract

Rain in bad weather becomes very challenging issue for automated driver assistance systems, security applications and many other computer vision algorithms. Number of rain removal algorithms have been proposed in these years but none of them has been able to increase the accuracy to large extent. In this paper a novel algorithm has been proposed where the slowly varying back ground scenes are taken for analysis. One part of the algorithm is based on the detection of rain pixels and the other part is removing them such that the original image must not lose its originality. Initially one background is modelled by taking the intensities of neighbourhoods of pixels. Then rain is detected by comparing each pixel with the modelled back ground followed by updating the background. The detected rain pixels are repaired with the help of some temporal neighbours. The results are promising to prove the work.

Subject Classification: *68T05 Learning and adaptive systems in artificial intelligence.*

Keywords: *Rain, Temporal correlation, Background modelling.*

1. Introduction

Removing rain drops in video is a tough task due to the random spatiotemporal distribution of rain and also the fast movement of rain. While removing rain drops from a video, at first it is required to detect rain drops correctly and then to remove it. The rain drop characteristics are being studied in atmospheric sciences [1-5]. As described in [6] rain is being detected using a probabilistic model with the help of some features

E-mail:* **bibhumohanty@soa.ac.in (Corresponding Author)

derived from rain analysis. In [9] a rain removal algorithm has been proposed which includes both temporal and chromatic properties of rain in video. A pixel does not contain rain drops throughout the video as per the temporal property of rain. Further change in R, G and B values in the rain affected pixels are nearly equal [9]. Using the above said property rain streaks can be detected and removed from static and dynamic videos whereas the videos are assumed to be taken from stationary cameras. Both light and heavy rain conditions have been analysed in their algorithm. According to Garg and Nayar [10] rain drops are spatially distributed, fall at high velocities and due to its reflection and refraction a sharp change in intensity is observed. Also rain drops undergo through motion blur due to time exposure of camera parameters. So they combined the dynamics of rain and rain photometry. In [11], a method of detection of rain has been explained where they have addressed only in the irradiance light field requiring three consecutive frames for computing temporal correlation of rain. Bernum et. al. [12] proposed a physical model of rain drops for determining the general shape and brightness of a single streak. They have analysed the statistical properties of rain in frequency domain, so as to supress the detected frequency components. In [14] a linear model for rain streaks has been proposed considering two factors; one, the intensity of the candidate pixel and the other is the background intensity. In the detection process, the neighbouring non rain pixels plays a vital role because the intensity variation between rain pixels and adjacent non rain pixels is significant. In [15] Sumit Kumar et. al. considered the intensity difference as the only parameter to detect rain streaks. A method for removing rain from a single image has been proposed in [16] where the gradient magnitude and direction of rain streaks are taken utmost care of. In this model, the rain pixels are considered to be of high frequency components which is removed by using a rain mask. The single image rain streak removal using depth of field and sparse coding method have been proposed by Xiao et. al. [17]. According to them the image has been decomposed with the combination of Short Time Fourier Transform (STFT) and bilateral filtering. Then the depth of the field saliency map of the image is being used to reduce rain residue in low frequency components and miss-matching is ignored. Another rain removal algorithm to remove rain from dynamic back ground videos has been proposed where the algorithm comprises of motion segmentation of dynamic scene [18]. Here the photometric and chromatic constraints are being used for rain detection. To remove rain particles a filtering approach is adopted by considering the motion occlusion in both temporal and spatiotemporal directions. A nonlinear model for de raining the rain affected single image frame has been proposed by Li et. al. [19]. For nonlinear rainy image decomposition, the screen blend model of Photoshop is being used. U-Dense Net model for rain detection and residual dense block for rain removal is being used. A rain map model has been proposed by Park et. al. [20]. The rain map is calculated by performing difference operation among the adjacent frames. A low pass filtering in saturation channel of HSV color space is performed for refining rain map. Hakkim proposed a method of low rank recovery to remove rain from videos [21]. He has used the Lagrangian multipliers method to detect the rain particles and tried to solve it through convex optimization form. Samantaray et. al. [23] proposed a neighbourhood based noise cleaning method which depends on spatial neighbourhood of pixels. In our rain removal algorithm we have considered the spatial relationship between pixels for detecting rain pixels as it is quite obvious that the intensity of rain affected pixel increases in comparison with its spatial neighbourhoods. In [24] Hove proposed a method which says that even if some data in our data base are degraded, then also our prediction based on these degraded data are not going to be affected that much. Tadesse et. al. [25] proposed Fuzzy decision making variables approach with the help of a mathematical example for decision making related issues. Using this decision making approach we can also try to predict a pixel being rain or non-rain but we need to model a rain drop first which is a difficult task. S Gupta et. al. [26] proposed the least square support vector machine algorithm for software maintainability prediction with the help of some data sets. Also we can predict the rain and non-rain pixels with the same support vector machine if we would have a good data set for real time analysis. A.K. Dubey et. al. [26] tried to find out a face recognition method where 1000 classes are there with a available data set. In our approach though we have only two classes but we do not have a data base which can help us in identification process. So the classifier based approach may not be suitable for rain detection algorithm.

2. Rain Characteristics

Water droplets which moves in high velocities are collectively known as rain. These droplets are random in nature. As per the detailed discussion in various atmospheric sciences regarding the characteristics of rain drops, the diameter of each droplet varies from 0.1mm to 3.5 mm [1-5]. According to Marshall-Palmer distribution [3], the drop size varies or increases exponentially and hence the density of rain drops decrease exponentially with the increase in radius. Initially the drop appears spherical and gradually it becomes a spheroids. In [6] the size of the rain drop is assumed to be of uniform. It was concluded from the experiment that the rain drop appears to be a streak and it lasts for a maximum of five consecutive frames in a video sequences.

The rest of the paper is organized as follows: the background extraction and modelling is discussed in Section 3. In Section 4, the methodology adopted for rain detection is discussed thoroughly; the technique for rain drop removal from the scene with model up gradation is discussed in Section 5; the methodology for ground truth generation from rain affected video is detailed in Section 6. Result and analysis of our proposed algorithm and future direction for the improvisation of the proposed algorithm as conclusion is discussed in Section 7 and Section 8 respectively.

3. Back Ground Extraction and Modelling

In this piece of work, we have assumed the shape and size of the rain drop to be uniform throughout the frames under consideration. Also it is assumed that the rain drops move in a uniform speed and it lasts up to three frames for better visual perception.

We have taken a videos where the back ground is varying slowly. So it is trivial that the back ground is assumed almost static and the rain drops create a positive fluctuation of intensity about mean over the image. The method of background extraction from a slowly varying background or static background video essentially requires the appearance of dependencies in consecutive rain pixels [7].

Let us consider N nos of frames from such a slowly varying back ground video. So the extracted background (BG) will be

$$BG = f(x,y) = median(v(x,y,i)) \tag{1}$$

$$where\ i = 1,2,3\ldots N,\ \forall(x,y).$$

Here $f(x, y)$ is the extracted background image without rain from the N initial frames and v is the initial stack of frames. Figure 1 shows one of the initial rain affected frames from the video sequences and the corresponding extracted back ground for set of frames.

It is a well-known fact that the neighbouring pixels share similar information and hence their statistical correlation do not vary significantly. Consider $m \times m$ neighbourhood of a pixel at (x, y) from BG.

$$R = \{r_1, r_2, r_3, \ldots r_n\} \tag{2}$$

Where r_i, i=1,2,3..n are the pixels from the $m \times m$ neighbourhood of location (x, y) and the pixels of r_i are arranged in descending order. So the background can be modelled as:

$$g(x, y, 1 : m \times m) = r_i, \forall i \tag{3}$$

Here g represents the newly formed stack of frames of size $(x \times y \times (m \times m))$ which we refer to as initial back ground model.

4. Rain Detection

As we have seen in Section 3, we now have $\boldsymbol{g(x, y, m \times m)}$ which we may refer as stack of back ground modelled frames. As rain gives positive fluctuations in intensity, the pixels will definitely suffer increase in intensity in comparison to its neighbours.

To distinguish between whether a pixel is rain affected or not, a comparison is made satisfying an empirically chosen but fact driven threshold value, say T1. It can be inferred from various literature that the significant change in intensity value will result if the difference in intensity of a pixel and its surrounding pixels must be around 15-20 percent of maximum grey scale intensity value of a pixel.

Hence,

C=a pixel is classified to be rain affected

$$= | current(x, y) - g(x, y, m \times m) | > T1 \tag{4}$$

and

$$Count(C) > T2 \ \ for \, \forall (x, y) \tag{5}$$

Where current (x, y) is the pixel under test at location (x, y) and $\boldsymbol{g(x, y, m \times m)}$ is the back ground frames as derived from equation (3). T2 is the threshold value which consider the fact that if 40-50 percentage of neighbouring pixel values are different then no need of any further counting and that pixel can be classified as rain pixels. Figure (2) shows the rain affected pixel and detected rain mask of frame no 101 from “Highway” video and frame no 47 from “Canoe” video.

5. Model Up gradation and rain removal

It is evident from the previous section, that equation (5) distinguishes between a pixel whether it is rain affected or not. Obviously if the pixel is not affected by rain i.e. if $Count(C) < T2$, then the same pixel can be considered as background pixel. If we want to update the frame we need to update the current pixel with any of the set of pixel from the pixel stack of the same location which will be helpful in rain detection from next frame by modelling as described in equation (6).

$$g(x, y, \boldsymbol{r}_i) = current(x, y) \quad \textit{for any random } i \tag{6}$$

As the pixel stack contains many pixels so any incorrect information which could wrongly inserted to the neighbourhood stack model does not affect the accuracy of the algorithm to large extent. The accuracy of the algorithm is largely affected if any sudden change in back ground creates a remarkable change in illumination and reflectance components. Also the accuracy gets hampered if the back ground contains significant illumination components having more illumination than rain. As explained in [6], the variation in intensity produced by the rain drops is symmetric about the mean. If we remove the affected pixels directly, then there is a possibility of leaving behind a black patches in the image. This results a distorted image. Hence removing the affected pixel may not be the right way in this case. Looking at the symmetry nature of the rain pixels, the following equation (7) is used for supressing the high intensity rain pixels.

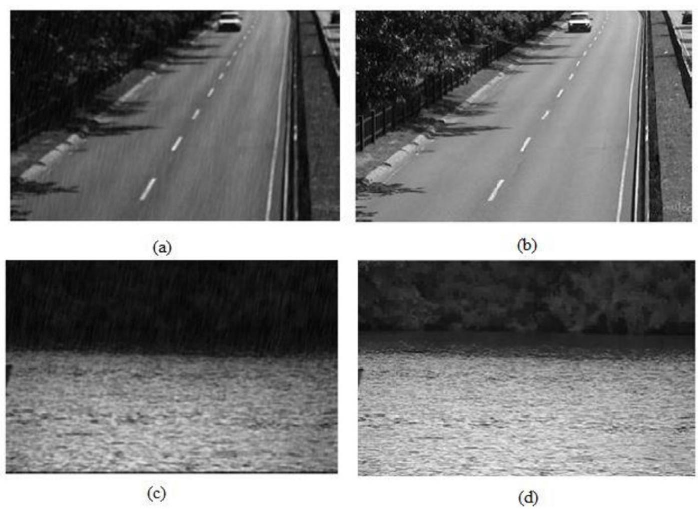

Figure 1

(a) rain affected frame (no-30) from video "High way" (b) corresponding back ground (c) rain affected frame (no-35) from video "Canoe" (d) corresponding back ground

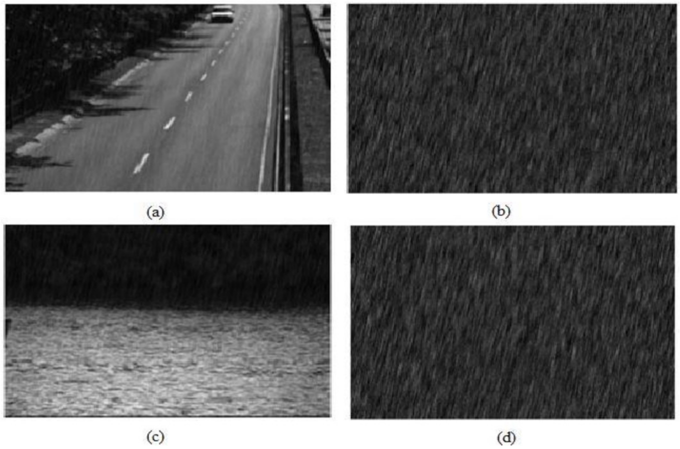

Figure 2

(a) rain affected frame (no-31) from video "High way" (b) corresponding detected rain mask (c) rain affected frame from video "Canoe"(no-36) (d) corresponding detected rain mask

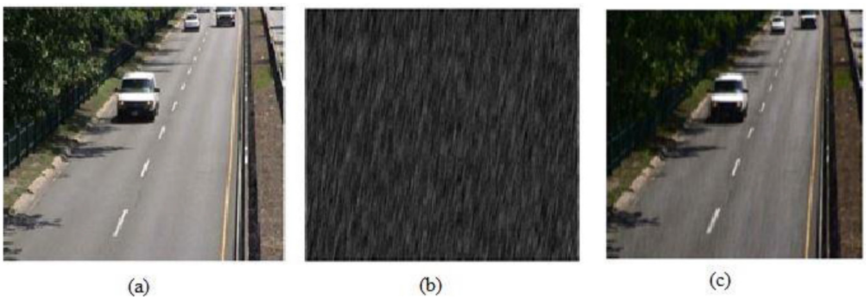

Figure 3

(a) original frame (b) rain mask (c) rain rendered frame

$$h(x,y)=\frac{1}{P}\sum_{z=z-\left(\frac{P}{2}\right)}^{z=z+\left(\frac{P}{2}\right)}H(x,y,z) \tag{7}$$

Where $h(x, y)$ is the intensity of location (x, y) after rain component is supressed, $H(x,y)$ is the rain affected frame and z is the temporal location.

Pseudo Code:

1. Choose a video sequence $\{V(\ )\}$

such that $\{V(\ )\} = v(x, y, z)$

for $x \epsilon X$, $y \epsilon Y$ and $z \epsilon Z$, $x \times y$ is the spatial dimension of the video frame and Z is the no of frame.

2. Select the initial N nos of original frames for generating background-neighbourhood planes:
   (a) $\{F()\} = v(x, y, N)$, for all $x \epsilon X$, $y \epsilon Y$
   (b) Applying median filter of length N at each spatial location in the temporal direction, such that the filtered frame f_m is created, i.e.

$$\begin{aligned}\{F_{med}()\} &= f_m(x, y, 1) \\ &= \{v(x, y, med(N))\}\end{aligned}$$

   for all $x \epsilon X$, $y \epsilon Y$

   The dimension of $F_{med}(\ )$ is $x \times y$
   (c) Select a neighbourhood size of $m \times m$ at a pixel location (x, y) in the $\{F_{med}(\ )\}$
      (i) For each (x, y) on $\{F_{med}(\ )\}$, sort its neighbourhood pixels in descending order of magnitude starting from the left hand corner.
      (ii) If (x, y) happens to be at the boundary of $\{F_{med}(\ )\}$, then duplicate the pixel values symmetrically around (x, y).
   (d) The BG-NH (background – neighbourhood) planes will be generated such that $R_{BG-NG}(x, y, \gamma_k) = f_{med}(x, y, k_k)$, the plane considering k^{th} neighbouring pixel of (x, y), where $k = 1, 2, 3 \ldots (m \times m)$
3. For any other rain affected frame $v(x, y, z)$, such that $z = N + 1, N + 2, ..z$,

   $C(x, y, z) = v(x, y, z')$ such that $z' = N + 1, N + 2, ..z$
4. If $|C(x, y, z) - R_{BG-NG}(x, y, \gamma_k)| > T_1$

   then $C_R = C_R + 1$,

   further if $C_R > T_2$ then $C(x, y, z)$ is the rain affected pixel

   else $R_{BG-NG}(x, y, \gamma_k) = C(x, y, z)$ is a background pixel
5. For each detected rain pixels in a frame of video sequence $v(x, y, z)$

$V_{out} = Avg(x, y, N-k : N-k)$, where (x, y) represents the rain detected pixel locations, N is the frame where we want to detect rain and k is empirically decided

6. Result validation

To validate the performance of our algorithm we need to create the ground truth which is a crucial and important stage in rain detection algorithm. Unless we do not know the ground truth of the video frames it is impossible in our part to estimate the performance of our algorithm. To create the ground truth it is necessary to create the rain mask, then to render the rain mask suitably in the video frames. In [7] Sonia. et. al. proposed a method of rain mask generation for static back ground videos. But in our case the video frames are slowly varying. So to generate the rain mask and ground truth we have adopted the rain rendering method explained in [8].Individual rain masks need to be generated for each video frames by assigning wind velocity and rain density. After generation they need to be rendered suitably to the frames to create a realistic rain appeared video. To do that, all the frames are converted to YIQ plane. The reason behind this is that we need to work in intensity plane only and the chrominance components are later added for good visual representations. Then it is being normalized and power law transformation is applied to darken it as explained in equation (8).

$$s = cr^{\gamma} \tag{8}$$

Where s represents the output intensity value corresponding to input intensity r with a suitably chosen γ value which is 7 for better visual effect. Then rain masks are created by assigning suitable velocities to rain drops and assigning suitable sizes. In this way the rain masks are created and then they are rendered to the darkened frames as explained in equation (9) and equation (10)

$$X = Y + Z \tag{9}$$

Where X represents the output intensity plane after addition of rain, Y represents the background and Z represents the rain mask for the frame which is calculated as per the following equation.

$$Z = \alpha(K - \beta Y) \tag{10}$$

Where K is the normalized, generated blurred mask. The value of $\beta \dot{o}$ (0,0.039) and $\alpha \dot{o}(0, 0.5)$. Figure 3 represents the rain masks and the rain affected frames as explained in this section.

7. Result and Analysis

In our algorithm the simulation is done in two videos with slowly varying background. One is "High way" video and the other is "Canoe" video. These two videos are taken with an assumption that the back ground remain static for initial couple of frames and then starts varying slowly. We have simulated all the work in MATLAB R2016a environment. We have also analysed all the qualitative and quantitative performances and compared it with the existing method.

Initially rain is being added synthetically in the videos as explained in Section 6. The method we have adopted also fits suitable in other non-rain videos. Some examples of rain rendering in other fast varying back ground videos are shown in Figure 4. As explained in Section 3, initially we have taken fifteen frames to calculate the back ground from the rainy videos. Once back ground is calculated then rain detection starts.

All the work has been done in intensity plane. The videos taken for simulation contains 240 × 320 pixels per frame each. The rain candidate pixels are being detected from the first frame as per the method explained in Section 3. The threshold values T1 and T2 are fact driven threshold values selected empirically as described in Section 4. The model is being up graded after each frame is being considered for analysis.

After rain is being detected from a frame, the quantitative performance of the algorithm is calculated with the help of positive fraction [22] which states that the number of correctly classified pixels divided by total number of rain affected pixels. Table 1 shows a comparison of the proposed algorithm with the spatiotemporal algorithm [6] for two different videos. Table 1 shows the values of positive fraction for frame number '112' from "High way' video and frame number '115' from "Canoe" video which clearly shows that our algorithm gives better result. Also the positive fraction values of consecutive frames of two videos are shown in Figure 5.

The detected rain pixels are then being in painted in such a way that the original content of the image must not be de graded or blurred. The result shows qualitatively good for a value of P=4. Choosing higher values of P may degrade the image. Qualitative results for "Canoe" video is shown in Figure 6. In this video one frame 967 is shown for qualitative analysis. Figure 7 shows the qualitative analysis for "High way" video for a random frame no 149.

Figure 4

(a) original frame from football video (frame no-25) (b) rain rendered frame (c) original image (google) (d) rain rendered image

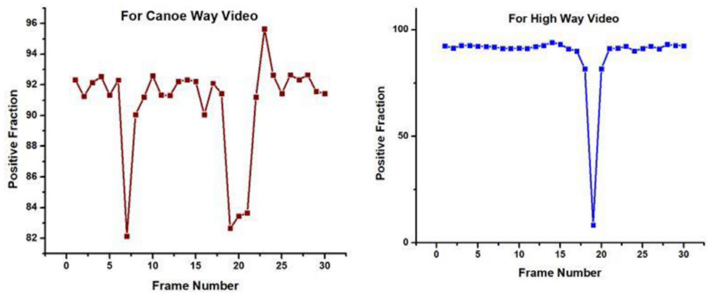

Figure 5

Positive fraction of a couple frames taken from Canoe and Highway video

Table 1

Positive fraction of two particular frames from two videos

	Positive Fraction in %	
Videos	**Tripathy et al method**	**Proposed Method**
Canoe	88.23%	91.26%
High way	85.91%	92.56%

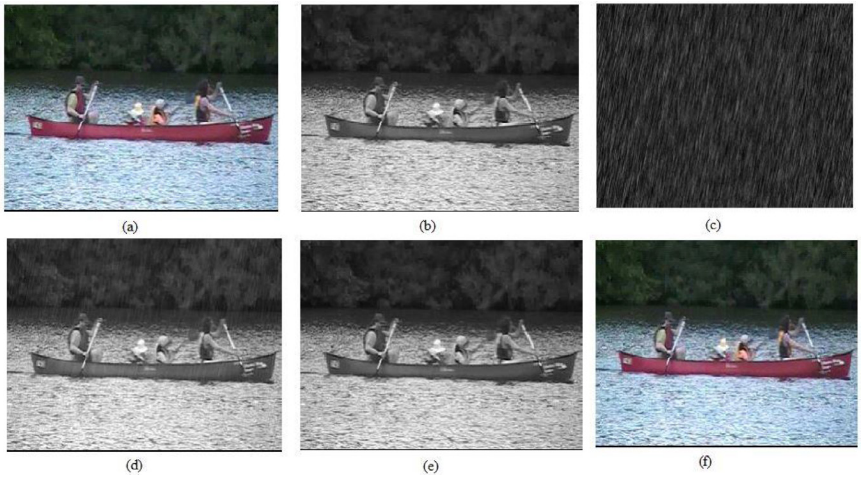

Figure 6

(a) frame no 967 from Canoe video (b) gray scale frame (c) rain mask for the frame (d) rain rendered gray scale frame (e) rain removed gray scale frame (f) rain removed RGB frame frame

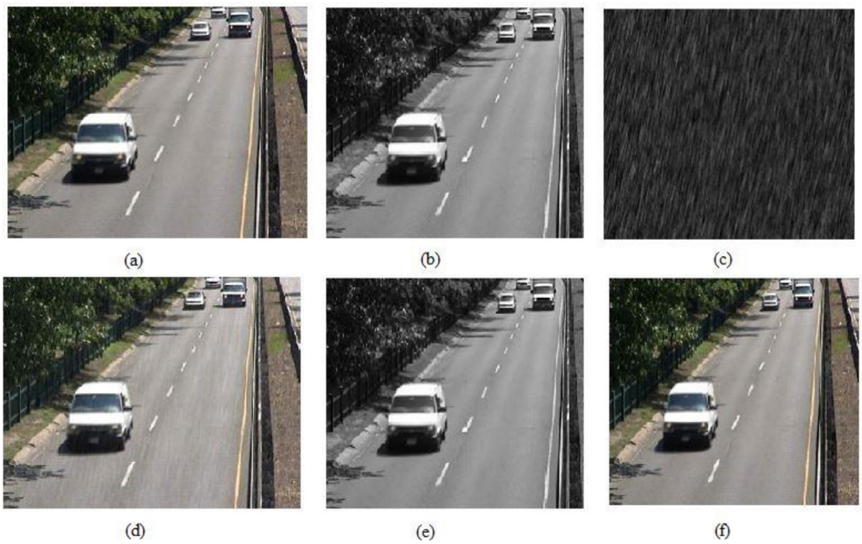

Figure 7

(a) frame no149 from High Way video (b) gray scale frame (c) rain mask for the frame (d) rain rendered gray scale frame (e) rain removed gray scale frame (f) rain removed RGB frame frame

8. Conclusion

In this paper, we have proposed a highly effective, novel rain removal algorithm. This algorithm works in intensity plane which makes it robust. We do not consider the velocity, droplet size of rain pixels. Also here the rain capturing camera is moving slowly as per the scene. Quantitative result shows the higher accuracy of the algorithm. As we are working on intensity plane, so the complexity of the algorithm is reduced. The rain removal algorithm in real time may help the driver in driving automated vehicles. The accuracy of the algorithm may be improved in future by considering more realistic situation.

References

[1] Gunn, R., Kinzer, G.D., "Terminal Velocity for Water Droplet in Stagnant Air" *Journal of Meteorology*, 1949, 6, pp.243-248.

[2] Beard, K.V., Chuang, C.H., "A New model for the equilibrium Shape of Rain Drops", *Journal of Atmospheric Science*, 1987, 44(11), pp.1509-1524.

[3] Marshall, J.S, Palmer, W.M.K,. "The Distribution of Raindrops with Sizes" *Journal of Meteorology* 1948, 5, pp.165–166.

[4] Manning, R.M., "Stochastic Electromagnetic Image Propagation", McGraw-Hill, Inc, 1993.

[5] Wang, T., Clifford, R.S., "Use of Rainfall-Induced Optical Scintillations to Measure Path-Averaged Rain Parameters", 1975, 8, pp:927–937.

[6] Tripathi, A. K. and Mukhopadhyay, S.,"A probabilistic Approach for Detection and Removal of Rain from Videos" , *IETE Journal of Research*, 2011, 57, pp:82-91.

[7] Starik, S. and Werman, m,: "Simulation of rain in videos" , Proceedings of Texture: The 3rd international workshop on Texture Analysis and Synthesis, France, 2003, pp.95-100.

[8] Tripathi,A.K. and Mukhopadhyay, S., "Rain Rendering in Videos", 4th International Conference On Computer Applications in Electrical Engineering Recent Advances, India, February 2010.

[9] Zhang, X., Li, H., Qi, Y., Kheng, L., Khim, T, "Rain Removal in video by combining temporal and chromatic properties", International Conference on Multimedia Computing and Systems, 2006.

[10] Garg, K., Nayar, S.K., "detection and removal of rain from videos", IEEE computer society conference on Computer vision and pattern Recognition, 2004, 1, pp:528-535.

[11] Shen, M., Xue, P., "A Fast Algorithm for rain detection and removal from videos", IEEE International Conference on Multimedia Computing and Systems, 2011.

[12] Barnum, P., Kanade, T., Narasimhan, S., "Spatio-Temporal Frequency Analysis for Removing Rain and Snow from Videos", Proceedings of Workshop on Photometric Analysis For Computer Vision (PACV), in conjunction with ICCV, 2007.

[13] Bahnsen, C., Moeslund, T., "Rain Removal in Traffic Survellance: Does it Matter?", IEEE Transaction on Intelligent Transportation Systems, 2019, 20(8), pp.2802-2819.

[14] Wang, Y., Liu, S., Xie, D., Zeng, B., "Removing Rain Streaks by Linear Model", ,IEEEAcess, 2020, 8, pp:54802-54815.

[15] Kumar, S., Gupta, A., Jha, R.K., "Analysis Diagnosis and Correction of Rain Streaks", TENCON-2019-2019 IEEE Region 10 Conference.

[16] Tang, H., Zhu, L., Zhang, D., Wang, X., "Single image rain removal using pure rain dictionary learning", IET Image Processing,1019, 13(10), pp:1797-1804.

[17] Xiao, J., Zou, W., Chen, Y., Wang, W., Lei, J., "Single image rain removal based on depth field and sparse coding", Pattern recognition Letters, 2018, 116, pp:212-217.

[18] Chen, J., Chau, L., "A Rain Pixel Recovery Algorithm for videos With Highly Dynamic Scenes", IEEE Transactions on Image Processing, 2014, 23(3), pp:1097-1104.

[19] Li, P., Tian, J.,Wang,G., Wu, C., "Model-Based Deep Network for Single Image Deraining", IEEE Access, 2020, 8, 14036 – 14047.

[20] Pak, K., Kim, M., Lim, H., Yu, S., Paik, J., "Fast rain Removl Using Rain Map and Temporal Filtering for Consumer Imaging Applications" IEEE 8th International Conference on Consumer Electronics 2018.

[21] E,Alaa, Hakim, A., "A Novel Approach for Rain Removal from Videos Using Low-Rank Recovery" 5th IEEE International Confference on Intelligent Systems, Modelling and Simulation. 2014.

[22] Prusty,P., Nayak, J., "Detection and Removal of Rain from satic Background videos", 2015, *International Journal of Applied Engineering and Research*,10(9), pp.22791-22797.

[23] Samantaray,A., Kanungo, P., Mohanty,B., "Neighborhood decision based impulse noise filter", 2018, IET Image Processing,12(7), pp.1222-1227.

[24] H.Hove, "Evaluation of statistical methods for assessing reliability from degradation data : A simulation study", *Journal of Statistics and Management Systems*, Vol 24, Issue 6, 2021.

[25] A. Tadesse, M. M. Acharya, M. Sahoo and S. Acharya "Fuzzy linear programming problem with fuzzy decision variables : A geometrical approach", *Journal of Statistics and Management Systems*, Vol 24, Issue 4, 2021.

[26] S. Gupta and A. Chug "Software maintainability prediction of open source datasets using least squares support vector machines", *Journal of Statistics and Management Systems*, Vol 23, 2020.

[27] A. K. Dubey and V. Jain "Automatic facial recognition using VGG16 based transfer learning model" , *Journal of Information and Optimization Sciences*, volume 41, Issue 7, 2020.