

# Visual Quality Enhancement Of Images Under Adverse Weather Conditions

1<sup>st</sup> Jashojit Mukherjee  
NVIDIA  
Bangalore, India  
jmukherjee@nvidia.com

2<sup>nd</sup> Praveen K  
NVIDIA  
Bangalore, India  
kpraveen@nvidia.com

3<sup>rd</sup> Venugopala Madumbu  
NVIDIA  
Bangalore, India  
vkrishnam@nvidia.com

**Abstract**—The visual quality of an image captured by vision systems can degrade significantly under adverse weather conditions. In this paper we propose a deep learning based solution to improve the visual quality of images captured under rainy and foggy circumstances, which are among the prominent and common weather conditions that attribute to bad image quality. Our convolutional neural network(CNN), NVDeHazeNet learns to predict both the original signal as well as the atmospheric light to finally restore image quality. It outperforms the existing state of the art methods by evaluation on both synthetic data as well as real world hazy images. The deraining CNN, NVDeRainNet shows similar performance on existing rain datasets as the state of the art. On natural rain images NVDeRainNet shows better than state of the art performance. We show the use of perceptual loss to improve the visual quality of results. These networks require considerable amount of data under adverse weather conditions and their respective ground truth for training. For this purpose we use a weather simulation framework to simulate synthetic rainy and foggy environments. This data is augmented with existing rain datasets to train the networks.

**Index Terms**—deraining, dehazing, cnn, deep learning

## I. INTRODUCTION

Visual quality degradation of images captured by vision systems under adverse weather conditions is an inherent problem spanning across multiple domains. There have been many techniques proposed in the past as a solution to this problem. Some solutions employed computer vision and image processing techniques [1]–[4], and recently there have been machine learning approaches [5]–[9] to solve this problem. In our paper, we propose a deep learning based solution for visual quality enhancement of images captured under rainy and foggy conditions which are among the most commonly observed weather adversities. The solution encompasses a deep neural network model and the data model used for the training the network. We propose two CNNs as a solution for deraining and defogging respectively in which we also optimize a perceptual loss function to improve the visual quality of the network output. For training the networks to perform deraining, the datasets Rain12, Rain100L and Rain100H have been used by Fu et al [10]. In our experiments we find the usage of this alone does not work the best in all kinds of scenarios and the dataset is somewhat limiting. So we use a weather simulation framework for generating rainy and foggy weather conditions in images to generate more data along with those preexisting datasets. The networks are



Fig. 1. Sample Comparison of Results (a) Rainy Image, (b) NVDeRainNet, (c) Foggy Image, (d) DehazeNet, (e) NVDeHazeNet

trained in a fully end to end manner without involving any other preprocessing step. The proposed solution can be used in the automotive domain for clearing up videos captured by vehicle mounted cameras, improving visibility and analytical potential of these images. This implies improving the accuracy of object detection or lane detection in images captured under poor weather. Also, this could be useful in various other domains including surveillance, military and sports.

Our primary contributions in this paper is to propose two CNNs for solving the task of single image rain streak removal and single image haze removal. The deraining network shows state of the art results on available datasets as well as on our augmented training data. It also outperforms existing methods in terms of qualitative and quantitative results on real world rainy images which is the primary goal of our work. We show through experiments that some network structures are better than others and also that usage of a perceptual loss function can improve the quality of images and reduce artifacts. Also, deeper or more complex network structures do not necessarily perform better which indicate the rain removal task is more of a spatially local problem closer to the pixel level than complex features. As the network is fully end to end there is no requirement to perform any image processing before or after processing the image using the network. The network is also fully convolutional which eliminates any constraints on the image size input to the network. The haze removal network also shows state of the art performance on the natural haze images. In this network we jointly predict

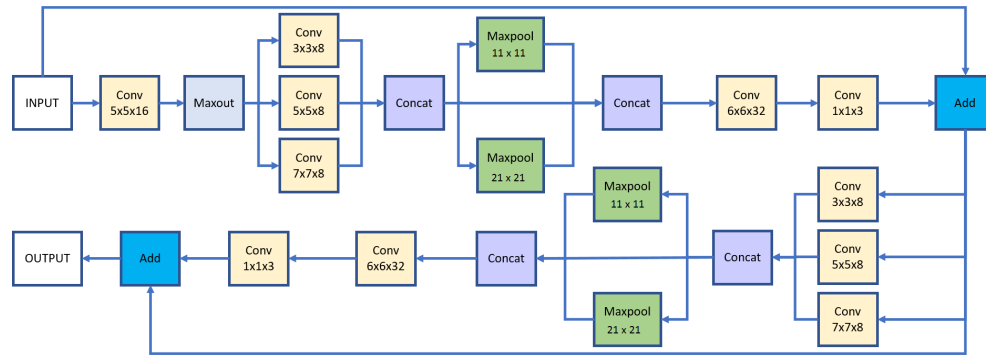


Fig. 2. NVDerainNet Network Architecture

the atmospheric light in images along with the original scene to perform better dehazing.

## II. RELATED WORK

A large number of methods have been proposed in literature to perform the task of removing rain streaks from images. Some methods have been proposed for rain removal in videos where the temporal information has been used [11]–[15]. Rain streak removal from a single image is a harder task. One class of algorithms approach this task as a layer separation problem. These try to separate the high frequency detail layer from a smooth base layer [1]–[3]. Then dictionary learning and sparse coding is used to remove the rain streaks. These methods suffer from the problem of over-smoothing the background, leading to loss of detail and blurring. Another approach detects rain streaks and uses a nonlocal mean filter to remove them [16]. Li et al [17] uses Gaussian mixture models and learned patch priors to remove rain streaks. Recently deep-learning approaches have been used to solve this task. Eigen et al. [7] use a deep learning approach to remove raindrops and dirt from images taken through a glass pane. The physical model for this problem is different from atmospheric rain and cannot be used for this task. Another recent deep learning based rain removal is introduced by Fu et al. [18] where the removal of rain streaks is done on the high frequency detail layer by learning a bank of filters. One drawback of the paper is that it is unclear when to use or not use image enhancement methods. In another paper, the same authors then introduce a new skip layer based network [10] which outperforms their earlier method. This method shows state of the art results and uses a Resnet like network architecture with 26 layers to remove rain streaks from the detail layer. The authors still perform a preprocessing step before passing the result as input to their network. In the paper by Yang et al. [5], a deep learning based method is introduced to effectively learn to jointly detect and remove rain from a single image.

In the case of haze removal too, extensive work has been done. For single image based haze removal, Tan introduced a technique [4] that maximizes local contrast in a hazy image with Markov Random Fields. Dark Channel Priors is another technique that has been used which exploits the fact that at least some pixels have low values for one of the channels in an image with no haze or fog [19]. This

is computationally intensive and also does not work for all types of images. Factorial MRF (Markov Random Field) based modeling [20] of the image have been used to increase image quality as well as to estimate the actual radiance of a scene better. Most of these methods are limited by the prior and heuristics they use which do not generalize to all kinds of scenes and images. Machine learning techniques have also been applied to this task. Zhu et al. [6] learns a supervised linear model to estimate the depth map of a hazy image. With the advent of deep learning many Convolutional Neural Network approaches have been used to handle similar problems like image super-resolution [8]. A recent work uses a CNN [9] to predict the medium transmission map of an image as proposed in the atmospheric scattering model. Then a sequence of pixel-wise operations is performed to obtain the haze free image.

Apart from specific methods for rain and haze removal there have been many recent developments which are relevant to solving this problem. Perceptual loss functions have been used to train the networks for problems like style transfer and super resolution by authors in multiple papers [21], [22]. This has shown success in improving the visual quality in image reconstruction tasks. To compute the perceptual loss, various Imagenet models can be used as they provide very good general representation of images. Due to its spatial resolution and good results VGG net is an ideal network for this task [23]. Also, Huang et al [24] in their paper introduce a densenet architecture for Imagenet where multi layer features are used to yield comparable results with much lesser number of neurons per layer.

## III. WEATHER MODEL

In this paper we consider two different kinds of weather conditions which adversely affect visual quality and thereby visibility. Rain, which manifests as streaks, reduces visibility and makes the image noisy. Also heavy rain or rain at a distance accumulate together to reduce visibility in a similar manner as fog or haze. This is the second form of weather adversity that causes very significant loss of visibility. Heavy fog can reduce visibility down to a few feet and make vehicle driving very difficult. This also adversely affects performance of computer vision algorithms that execute in the software stack of ADAS (Advanced driver-assistance systems).

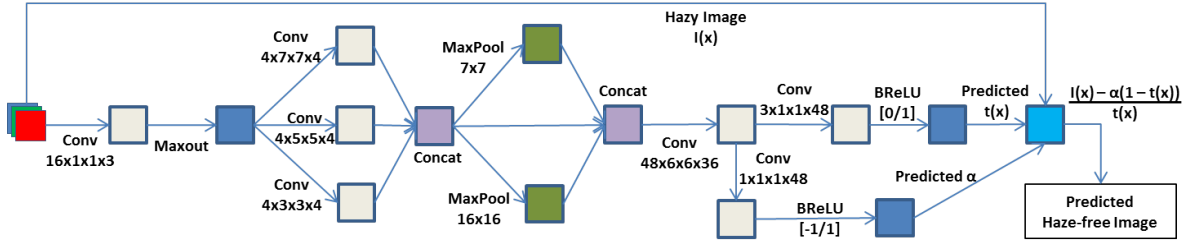


Fig. 3. NVDeHazeNet CNN architecture

#### A. Formation of Rain

Rain manifests itself as streaks in an image with varying density, opacity, length, direction as well as increasing amount of haziness as rain accumulates with depth. The streaks are similar to an additive component to the original image. As widely used in literature [3], [17], this can be written formally as

$$I(x) = J(x) + R(x) \quad (1)$$

Here  $I(x)$  is the observed image while  $J(x)$  is the original clear scene.  $R(x)$  is an additive component introduced as rain streaks for pixels  $x$  where a rain streak is present. The original color and detail information of the rain affected pixels are suppressed in  $I(x)$ .

#### B. Formation of Fog

Presence of fog increases atmospheric scattering of light which reduces visibility in images. This effect is more pronounced for objects at more distant depths as the scattering exponentially increases with depth. Following the work in [25], [26], the atmospheric scattering model can be formally expressed as:

$$I(x) = J(x) * t(x) + \alpha(1 - t(x)) \quad (2)$$

Where  $I(x)$  is the observed image and  $J(x)$  is the original clear scene.  $t(x)$  is known as the transmission medium map and  $\alpha$  is the global atmospheric light and  $x$  is the index for each pixel in an image. To obtain a clean image  $J(x)$  given  $I(x)$  we need to estimate the values of  $t(x)$  and  $\alpha$ . The value of  $t(x)$  is defined as

$$t(x) = e^{-\beta d(x)} \quad (3)$$

where  $\beta$  is the scattering coefficient and  $d(x)$  is the depth of the scene at pixel  $x$ . So the value of  $t(x)$  is bounded between 0 and 1 as depth is bounded between 0 and 1 with objects very near having value close to 0 and objects infinitely distant have depth value 1.

### IV. VISUAL QUALITY ENHANCEMENT USING CNN

We propose two end to end deep learning CNNs to solve the task of improving visibility in adverse conditions of rain and fog. The networks are created and trained in a fully convolutional manner such that there are no dependencies on image size. At each intermediate stage of the network the image size is kept constant and as a result there is no loss in resolution. This is mostly possible since after experiments we

reach the conclusion that the corruption caused by rain and fog in an image is mostly local in nature and a receptive field covering the largest rain streaks is enough to solve this low level task. For this reason we are able to train our networks with patching thereby increasing the amount of data available for training.

**Rain Removal Network:** Our rain removal network is inspired by the network proposed by Cai et al [9] as well as resnets. The network architecture is as shown in Figure 2s. The first stage of the network consists of an initial 5x5 convolution layer with 16 kernels followed by a maxout non-linearity. The max out layer simply outputs the maximum value at each image location across 4 groups of 4 layers on the result of the 5x5 convolution layer. The output filter map of 4 channels produced introduces a competing bottleneck which effectively learns a good non linear mapping of the original RGB image. This result is then fed into a bank of multi scale convolution layers consisting of 16 kernels each of 3x3, 5x5 and 7x7 filters. These simultaneously provides a larger receptive field while also providing more fine grained information. Max pooling with two different window sizes of 11x11 and 21x21 is performed to accommodate large rain streaks as well as multiple scales. The result of the pooling layers and the previous convolution layer output is concatenated before being passed to another convolution layer of size 6x6 and consisting of 128 kernels. The result of this layer is finally passed into a 1x1 non linear mapping layer which maps it to a layer of 3 channels. This layer is bounded between values -1 and +1 and added to the original image to generate the output of the first stage. The next stage repeats the previous operations much like a resnet with skip connection to obtain the final cleaned up image. Padding is used to maintain the original size of the image in every layer. Figure 2 shows the architecture of our final network. The non linearities applied after each convolution layer are not shown in this figure. This network has 171928 parameters and 134 neurons. **Haze Removal Network:** Our haze removal network is also inspired by Cai et al [9] although it has significant differences. The first convolutional layer kernel size is set to 1x1 to make it into a nonlinear mapping kernel. This maps the RGB values of each pixel to 16 different weighted combinations which are learned. The next two layers are similar to what we use in the deraining model with two max-pooling layers. Two further convolutional layers are then added to produce a 3 channel transmission map. Unlike DehazeNet we not only learn the transmission map

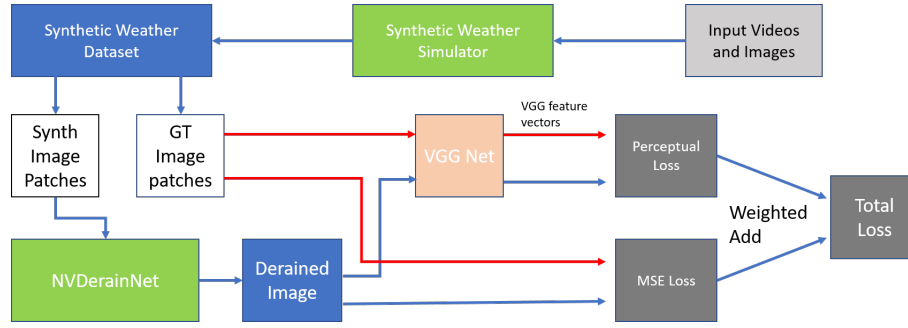


Fig. 4. Training Block Diagram

but also the atmospheric light value,  $\alpha$ . This is done per pixel with another convolutional layer. This removes the need to manually compute the value of global atmospheric light. Finally the haze free output image is computed by

$$J(x) = (I(x) - \alpha(1 - t(x)))/t(x) \quad (4)$$

This value is passed through a BReLU activation function introduced in [9] to set its bound between 0 and 1. Figure 3 shows the architecture of NVDehazeNet along with the intermediate filter map and kernel shapes.

## V. EXPERIMENTS

### A. Synthetic Data Generation

Acquiring training data for this problem poses significant challenges. We need the ground truth information for every piece of data we use to train the networks. This is nearly impossible to achieve without a synthetic data generation mechanism. Previous works which have tackled this problem have typically used an image editing software to add artificial rain streaks on images and use this as training data. Fu et al [10], in their paper uses such a dataset which is publicly available. However we found this dataset inadequately representing all kinds of rainy and foggy images. Hence, we use a completely configurable weather simulation framework to generate more synthetic training data. There are multiple challenges in accurately modeling these weather conditions. To begin with, we need an environment which is dynamic in order to achieve diversity in data. Automotive driving environments seemed suitable for this purpose. Next, we need a mechanism to simulate diverse weather conditions. One plausible method is to simulate the entire 3D environment with a fake world, but this approach is extremely challenging in terms of the realism that can be portrayed. Another method is to overlay the rain or fog effect on top of pre-rendered videos. This technique seems to be a practical solution to us, since there exists a lot of pre-rendered driving videos. The challenge of using this technique however, is to portray realism even though the rain or fog generated is actually not interacting with the background environment. Another challenge is that since we need to apply this effect over a large volume of data, performance is a key factor. Hence, leveraging more computation power by preferably using the GPU is always in contention. Last but not the least, we also need mechanisms to configure rain and fog variably to ensure

diverse data generation. The implementation of these models are based on techniques mentioned in [27], [28] and [29]. In [29], the authors propose a network which is able to predict the depth map from monocular images. When rendering the fog we use the predicted depth map using their model to use for the depth factor. In case such data is unavailable the depth factor is set to an uniform factor over the whole image. Both these models are used to augment combinations of rain and fog of varying inherent properties. This synthetic data and its corresponding ground truth are used for training the CNNs we have proposed in this paper.

### B. Dataset Creation

Since both rain and haze removal are relatively low level tasks we use patches from the training images to finally train our network. Several input videos of driving under cloudy conditions were downloaded from youtube and synthetic weather conditions were generated with the above framework. We focus on driving videos as automotive domain is one of the relevant areas in which deraining and dehazing can be applied. Around 4000 frames per video were used and upon adding different weather conditions a total of around 100000 frames were generated. From each frame 25 random 100x100 patches were selected along with the base ground truth thereby creating around 2.5 million image patches with varying weather conditions. Similar patching was performed on the previously available datasets like Rain12, Rain100L and Rain100H. This was further augmented with data from the SUN [30] and BSD500 [31] datasets. Synthetic rain was added on these images as with the driving videos and then patches were created in a similar manner for training. The increased diversity of backgrounds in the data show improvement in performance of networks.

A natural rain and fog dataset was also compiled for testing purposes. This contained real life images containing rain and fog without any synthetically generated weather condition. The performance of the networks on these images bear testimony to the effectiveness in solving the problem at hand. Some examples have been shown in figures.

### C. Network Training

We used Tensorflow [32] as the framework for implementing our CNNs. We train the networks on patches on size 100x100 with the assumption that the corruption due to rain and fog causes loss in information which can be recovered





Fig. 5. a) Source Image, b) Synthetic Rain, c) Deep Detail Net, d) NVDeRainNet

from the local neighborhood. Batch size used for the final networks was 96 and a learning rate of 0.0001 was used for training. The Adam optimizer was used to minimize the loss function. The network was trained for 400 epochs. As shown in Figure 4, two loss functions are used for training. Initially the training was done using only Mean Squared Error loss on the predicted image pixels with respect to the ground truth. However this method also introduces significant amounts of artifacts and the cleaned images are less visually appealing. For this purpose we use perceptual loss. We use a pretrained VGG19 network with frozen weights to compute the feature vectors of size 4096 just before the final classification layer. These vectors are a very good representation of images and hence the loss computed at this point with respect to the ground truth helps in improving the visual quality of the images. The loss is backpropagated through the vgg network without updating the weights of the vgg network. At the output layer of our network we now have two losses. One is the backpropagated error from the vgg network and the other is the mean squared loss computed at the pixel level. The relative amounts of the two losses to be backpropagated is controlled as a parameter and the weighted average of the losses is used to compute the gradients in our network. For our best network the weight used for MSE loss is 0.7 while the weight for the perceptual loss is 0.3. Table 2 demonstrates the performance metrics obtained on our validation set by training NVDerainNet with and without perceptual loss.

#### D. Network Variants and Observations

Initially for rain streak removal we experimented with both single frame and multi-frame rain streak removal techniques. Although multi-frame provided arguably better results, we abandoned that approach as it lacked generality. For single image case we trained several networks ranging from heavy networks with a large number of parameters which are slow to perform, and light fast smaller networks. Also different levels of fog were introduced to the images. Training with heavy amount of fog only brought in undesired artifacts in places of heavy fog with very little background data. This can be explained by the fact that the training data had very clean patches as ground truth and the foggy images had very little information to recover that clean patch. So

TABLE I  
COMPARISON OF METRICS ON VALIDATION SET FOR NETWORK VARIANTS

MultiScale Layer	MSE	NRMSE	PSNR	SSIM
8 kernels	0.001530	0.089270	29.609410	0.932060
16 kernels	0.001250	0.077220	30.887350	0.941230
32 kernels	0.001460	0.086480	30.411160	0.935800

TABLE II  
COMPARISON OF METRICS ON VALIDATION SET WITH AND WITHOUT PERCEPTUAL LOSS

Loss	MSE	NRMSE	PSNR	SSIM
MSE	0.00173	0.09573	29.25345	0.92893
MSE + Perceptual	0.001250	0.077220	30.887350	0.941230

the dataset was augmented by lighter fog with significant amount of information available in the pixels to reconstruct the background. Also receptive fields were increased as more contextual information was required to deal with heavy fog scenarios. To do this we tried adding more convolutional layers as well as adding a maxpooling of larger size. Another technique we tried was to use dilated convolutions but those provided inferior results in our preliminary experiments as compared to multiple filter sizes.

With the recent release of the DenseNet architecture for Imagenet we tried a dense net for the task. This dense network has two dense blocks each containing 5 convolution layers. Each convolution layer has 16 kernels of size 3x3 and the result is concatenated with the previous filters and fed into the next layer. The last convolution layer has 3 kernels to produce a 3 channel rain image which is added to the original image in the manner of a skip connection to produce the derained image. This is repeated in the second dense block to produce a final cleaned up image. As with earlier experiments this too benefits from introducing a perceptual loss factor along with the MSE loss. This showed good results although they were marginally inferior to NVDerainNet. This network also performs well for fog removal but is slower than NVDehazeNet. Table 1 shows the performance metrics on our validation set when varying the number of kernels in the multiscale convolution layers for NVDerainNet. Other variants including more different sized filters ie 1x1 and 11x11 filters showed inferior performance. A similar result

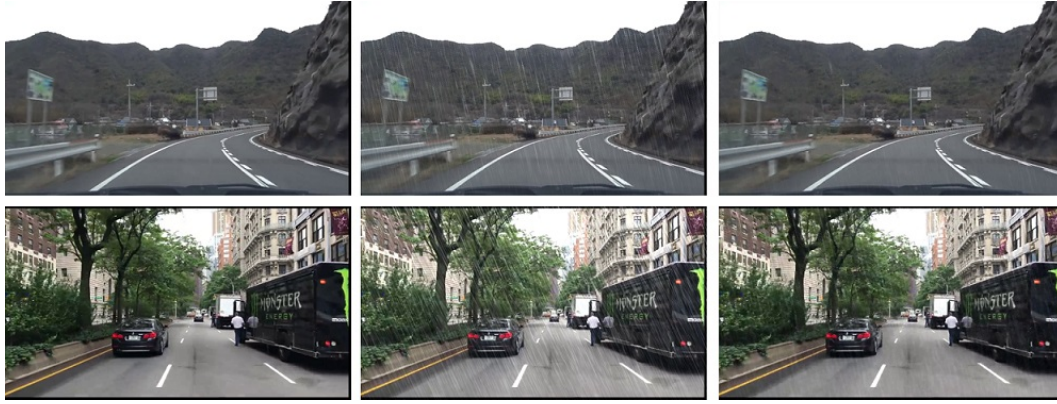


Fig. 6. a)Original Clean Images, b)Simulated Rain Images and c)NVDeRainNet

was obtained with the dense net architecture. The Structured Similarity Index Measure (SSIM) on the validation set for both these cases were around 0.92. One possible explanation for the failure of such complex networks is the fact that the task at hand is largely local in the spatial neighbourhood and very complex high level features are not required for this rain removal.

## VI. RESULTS

TABLE III  
COMPARISON WITH DEHAZENET FOR SYNTHETIC FOG

Haze	MSE	NRMSE	PSNR	SSIM	BIQA
Synthetic(fog)	5249.537	0.577	11.735	0.733	26.913
DehazeNet	787.118	0.204	21.613	0.855	24.531
NVDeHazeNet	783.461	0.217	20.048	0.841	21.816

TABLE IV  
COMPARISON WITH DEEP DETAIL NETWORK FOR SYNTHETIC RAIN DATA

Rain	PSNR	SSIM
Synthetic(rain)	22.0694	0.7063
Deep Detail Net	27.5288	0.8453
NVDeRainNet	27.5493	0.8545

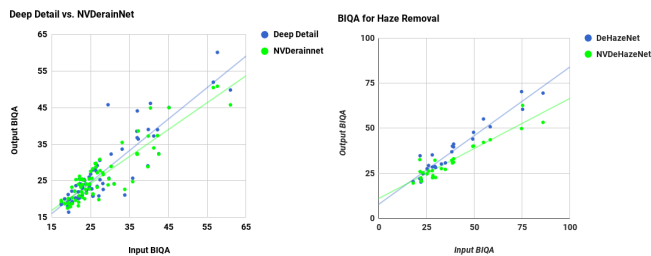


Fig. 7. (a) Rain Images (b) Fog Images

In our experiments we trained our network with the synthetic data generated by our weather generation framework. To test the effectiveness of our network we also perform a series of tests and compute several metrics that provide quantitative measure of the quality of the results. For this purpose we use both synthetic images from our generation framework and also more importantly natural images of

rain and haze. The results on these images reflect the true performance of our network in the real world. Although the network is trained with per pixel Mean Squared Loss this is not the best measure while comparing images. Also this kind of comparison is only possible with synthetic training data where a ground truth is known. We compute the mean Peak Signal-to-Noise Ratio as well as mean Structural Similarity Index for 96 haze images and 144 rain images with varying degree of visibility. These values are shown in Table 3 where it is compared with the results of DehazeNet [9] which claim state of the art performance. The same measures are also given for the synthetic data to just illustrate how badly image quality is affected by rain/fog. In Table 4, the results of our model are compared with the results of Deep Detail Network [10]. We studied other methods of deraining but due to lack of availability of code in some papers like Yang et al [5] does not allow us to directly compare the results. Since the Deep Detail network is state of the art we compare all our deraining results against it. Although the final image mean squared error is used to optimize the network, this is not a good metric to compare images. Far more robust and commonly used metrics are Peak Signal-to-Noise Ratio (PSNR) and Structural Similarity Index Metric (SSIM) [33]. Both these methods require a ground truth for comparison and a higher value indicates a better result. While the PSNR deals with how much the noisy component has reduced in an image it does not deal with the contents of the image. In that regard the SSIM is a better metric as it finds the structural similarity between two images. As seen in table 4 our model performs better than Deep Detail Network in terms of PSNR and SSIM on the synthetic data. Our de-raining network also successfully manages to remove slight haze in the image as demonstrated by the buddha image (Figure 10). Fu et al [10] in their paper show structural similarity on a subset of images from their training data. We randomly select 500 images from their dataset and compare the results with ours in table. We achieve slightly higher PSNR and a better SSIM score on these images.

For natural images of rain and fog there is no reference ground truth and so the above measures cannot be applied. Here we use Blind Image Quality Assessment(BIQA) [34]measure to get an estimate about the visual quality of a



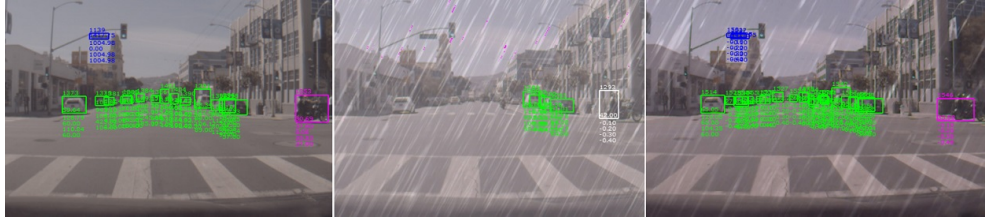


Fig. 8. Object Detection Results in a)Original Clean Images, b)Simulated Rain Images and c)NVDeRainNet

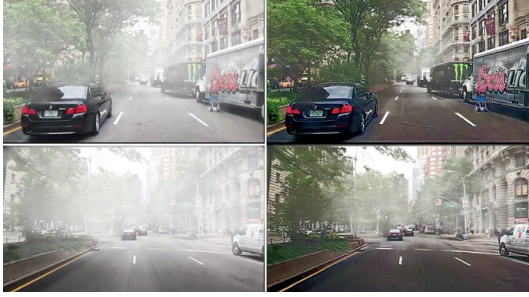


Fig. 9. Defogging Images

TABLE V  
COMPARISON OF MEAN BIQA FOR NATURAL IMAGES

Rain	BIQA	Haze	BIQA
Rainy Image	27.8460	Hazy Image	38.294
Deep Detail Net	27.1167	DehazeNet	36.941
NVDeRainNet	26.3980	NVDeHazeNet	32.257

stand alone image. Similar comparisons with DehazeNet and DerainNet are provided using this measure for 25 natural haze and 75 natural rain images in table 5. Our results look visually much better than the compared papers and this is quantified with BIQA. The mean BIQA index for the natural haze images with NVDeHazeNet is significantly better than DehazeNet. For natural rain images the mean BIQA index with NVDeRainNet is better than Deep Detail Network. Figure 7 Graphs (a) and (b) plot the input and output BIQA index for each natural rain and haze image. We find that output image BIQA index for our network in both cases (red line) is better than the trend for the outputs from comparison networks. Also for heavy fog which produces images with very high BIQA index, as seen on the far right of graph (b), our network significantly outperforms DehazeNet.

In case of DehazeNet our fog model produces comparable if slightly inferior results in terms of PSNR and SSIM (Table 3). However our BIQA score for these images are better than DehazeNet. To test for robustness we also check how our network performs when it is presented with images which are taken in perfectly normal weather conditions. No significant artifacts or distortion was found in such cases. We also performed experiments to demonstrate the usefulness of NVDeRainNet as a preprocessing step to other ADAS use-cases. The synthetic weather simulator was used to add rain on a labeled video for object detection. We use the same object detection network to detect objects like cars and pedestrians for three cases: a) original video, b) the same video with simulated rain and c) the rainy video cleaned up by our

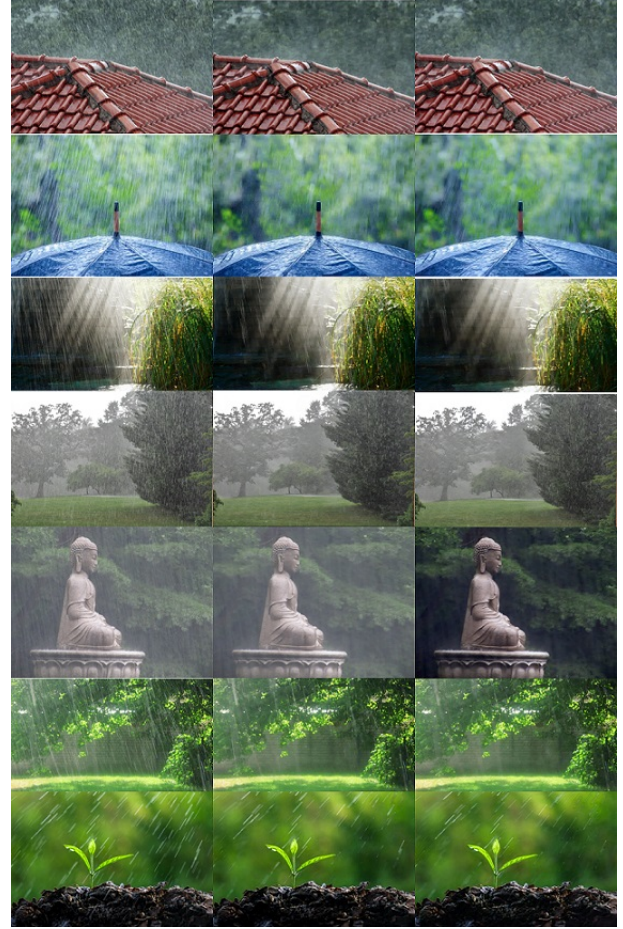


Fig. 10. a) Natural Rain Images, b) Deep Detail Net, c)NVDeRainNet

network. The images in the figure 8 demonstrate the results for this experiment. The recall for all objects in the original video was 0.41 using the object detection network. For the rainy video this dramatically falls to 0.043 demonstrating the degradation of performance in such weather conditions. Upon cleaning up the recall improves to 0.342.

## VII. CONCLUSION AND FUTURE WORK

In this paper we propose CNN based solutions for Haze and Rain removal from images. For haze removal the key concepts introduced are to learn the atmospheric light for each pixel as well as the transmission map. For the rain removal network we introduce the use of perceptual loss in combination with MSE loss to produce better results. To help train these networks we use a synthetic weather simulation

framework which can be configured to simulate varying kinds of rain and haze details. Our method outperforms the other methods in terms of mean SSIM and PSNR on synthetic data as well as on mean BIQA index for natural rain or haze images. This paper introduces two different networks to handle the cases of rain and haze. In the future a single network can be trained which can produce clean images regardless of which weather aspect causes the loss of visibility. It can also be extended to other weather conditions and physical models. Also a lighter network which can be used as a preprocessing step in a real-time application like object detection or lane detection can be potentially trained. We consider these as future work.

## REFERENCES

- [1] D. A. Huang, L. W. Kang, Y. C. F. Wang, and C. W. Lin, "Self-learning based image decomposition with applications to single image denoising," *IEEE Transactions on Multimedia*, vol. 16, no. 1, p. 8393, 2014.
- [2] L. W. Kang, C. W. Lin, and Y. H. Fu, "Automatic single image-based rain streaks removal via image decomposition," *IEEE Transactions on Image Processing*, vol. 21, no. 4, p. 17421755, 2012.
- [3] Y. Luo, Y. Xu, and H. Ji, "Removing rain from a single image via discriminative sparse coding," *International Conference on Computer Vision (ICCV)*, 2015.
- [4] R. T. Tan, "Visibility in bad weather from a single image," *International Conference on Computer Vision and Pattern Recognition (CVPR)*, pp. 1–8, 2008.
- [5] W. Yang, R. T. Tan, J. Feng, J. Liu, Z. Guo, and S. Yan, "Joint rain detection and removal via iterative region dependent multi-task learning," *CoRR*, vol. abs/1609.07769, 2016. [Online]. Available: <http://arxiv.org/abs/1609.07769>
- [6] Q. Zhu, J. Mai, and L. Shao, "A fast single image haze removal algorithm using color attenuation prior," *International Journal of Computer Vision*, vol. 98, no. 3, p. 263278, 2012.
- [7] D. Eigen, D. Krishnan, and R. Fergus, "Restoring an image taken through a window covered with dirt or rain," *International Conference on Computer Vision (ICCV)*, 2013.
- [8] C. Dong, C. C. Loy, K. He, and X. Tang, "Image super-resolution using deep convolutional networks," *IEEE Trans. Pattern Anal. Mach. Intell.*, vol. 38, no. 2, p. 295307, 2016.
- [9] B. Cai, X. Xu, K. Jia, C. Qing, and D. Tao, "Dehazenet: An end-to-end system for single image haze removal," *IEEE TRANSACTIONS ON IMAGE PROCESSING*, vol. 25, no. 11, 2016.
- [10] X. Fu, J. Huang, D. Zeng, Y. Huang, X. Ding, and J. Paisley, "Removing rain from single images via a deep detail network," in *The IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*, July 2017.
- [11] K. Garg and S. K. Nayar, "Detection and removal of rain from videos," *International Conference on Computer Vision and Pattern Recognition (CVPR)*, 2004.
- [12] P. C. Barnum, S. Narasimhan, and T. Kanade, "Analysis of rain and snow in frequency space," *International Journal on Computer Vision*, vol. 86, no. 2-3, pp. 256–274, 2010.
- [13] J. Bossu, N. Hautiere, and J. Tarel, "Rain or snow detection in image sequences through use of a histogram of orientation of streaks," *International Journal on Computer Vision*, vol. 93, no. 3, p. 348367, 2011.
- [14] V. Santhaseelan and V. K. Asari, "Utilizing local phase information to remove rain from video," *International Journal on Computer Vision*, vol. 112, no. 1, p. 7189, 2015.
- [15] N. Brewer and N. Liu, "Using the shape characteristics of rain to identify and remove rain from video," *Joint IAPR International Workshops on SPR and SSPR*, p. 451458, 2008.
- [16] J. H. Kim, C. Lee, J. Y. Sim, and C. S. Kim, "Single-image deraining using an adaptive nonlocal means filter," *IEEE International Conference on Image Processing (ICIP)*, 2013.
- [17] Y. Li, R. T. Tan, X. Guo, J. Lu, and M. S. Brown, "Rain streak removal using layer priors," *International Conference on Computer Vision and Pattern Recognition (CVPR)*, p. 27362744, 2016.
- [18] X. Fu, J. Huang, X. Ding, Y. Liao, and J. Paisley, "Clearing the skies: A deep network architecture for single-image rain removal," *CoRR*, vol. abs/1609.02087, 2016. [Online]. Available: <http://arxiv.org/abs/1609.02087>
- [19] K. He, J. Sun, and X. Tang, "Single image haze removal using dark channel prior," *IEEE Trans. Pattern Anal. Mach. Intell.*, vol. 33, no. 12, p. 23412353, 2011.
- [20] K. Nishino, L. Kratz, and S. Lombardi, "Bayesian defogging," *Int. J. Comput. Vis.*, vol. 98, no. 3, p. 263278, 2012.
- [21] P. Isola, J.-Y. Zhu, T. Zhou, and A. A. Efros, "Image-to-image translation with conditional adversarial networks," *CoRR*, 2016. [Online]. Available: <http://arxiv.org/abs/1611.07004>
- [22] C. Ledig, L. Theis, F. Huszar, J. Caballero, A. Cunningham, A. Acosta, A. Aitken, A. Tejan, J. Totz, and Z. W. et al, "Photo-realistic single image super-resolution using a generative adversarial network," *CoRR*, 2016. [Online]. Available: <http://arxiv.org/abs/1609.0480>
- [23] K. Simonyan and A. Zisserman, "Very deep convolutional networks for large scale image recognition," *CoRR*, vol. abs/1409.1556, 2014. [Online]. Available: <http://arxiv.org/abs/1409.1556>
- [24] G. Huang, Z. Liu, and K. Q. Weinberger, "Densely connected convolutional networks," *CoRR*, vol. abs/1608.06993, 2016. [Online]. Available: <http://arxiv.org/abs/1608.06993>
- [25] E. J. McCartney, "Optics of the atmosphere: Scattering by molecules and particles," Wiley, vol. 1, 1976.
- [26] S. K. Nayar and S. G. Narasimhan, "Vision in bad weather," *IEEE Conf. Computer Vision*, vol. 2, p. 820827, 1999.
- [27] D. Hoskins, "Rain shader," <https://www.shadertoy.com/view/XdSGDc>, 2014.
- [28] S. Craitoiu, "Fog shader," <http://in2gpu.com/2014/07/22/create-fog-shader/>, 2014.
- [29] D. Eigen, D. Krishnan, and R. Fergus, "Depth map prediction from a single image using a multi-scale deep network," *Advances in Neural Information Processing Systems*, 2014.
- [30] J. Xiao, K. A. Ehinger, J. Hays, A. Torralba, and A. Oliva, "Sun database: Exploring a large collection of scene categories," *Int. J. Comput. Vision*, vol. 119, no. 1, pp. 3–22, aug 2016. [Online]. Available: <http://dx.doi.org/10.1007/s11263-014-0748-y>
- [31] D. Martin, C. Fowlkes, D. Tal, and J. Malik, "A database of human segmented natural images and its application to evaluating segmentation algorithms and measuring ecological statistics," in *Proc. 8th Int'l Conf. Computer Vision*, vol. 2, July 2001, pp. 416–423.
- [32] M. Abadi, A. Agarwal, P. Barham, E. Brevdo, Z. Chen, C. Citro, G. S. Corrado, A. Davis, J. Dean, M. Devin, S. Ghemawat, I. J. Goodfellow, A. Harp, G. Irving, M. Isard, Y. Jia, R. Józefowicz, L. Kaiser, M. Kudlur, J. Levenberg, D. Mané, R. Monga, S. Moore, D. G. Murray, C. Olah, M. Schuster, J. Shlens, B. Steiner, I. Sutskever, K. Talwar, P. A. Tucker, V. Vanhoucke, V. Vasudevan, F. B. Viégas, O. Vinyals, P. Warden, M. Wattenberg, M. Wicke, Y. Yu, and X. Zheng, "Tensorflow: Large-scale machine learning on heterogeneous distributed systems," *CoRR*, vol. abs/1603.04467, 2016.
- [33] Z. Wang, A. C. Bovik, H. R. Sheikh, and E. P. Simoncelli, "Image quality assessment: From error visibility to structural similarity," *IEEE TRANSACTIONS ON IMAGE PROCESSING*, vol. 13, no. 4, pp. 600–612, 2004.
- [34] L. Zhang, L. Zhang, and A. C. Bovik, "A feature-enriched completely blind image quality evaluator," *IEEE Trans. Image Processing*, vol. 24, no. 8, p. 25792591, 2015.