探究attentive GAN的掩膜矩阵和自编码器损失函数的实验

Huixiang Luo[1]
[1] Fudan University
15307130012@fudan.edu.cn

Abstract

当我们在雨中拍摄照片的时候，由于雨滴附着在玻璃上、镜头上而造成图像的某些区域变得模糊和失真是很让人头疼的事情。如何去除这些雨滴呢？目前，基于深度学习的方法在图像去雨滴领域取得了较好的成果。基于 *GAN*网络和注意力机制的*attentive GAN*是一个自动化的图像去雨滴架构，它的去雨滴图像可视化效果非常好。本文探究了它的内部结构，并尝试着改动了其中一部分代码，在定量指标上取得了一定的提升。

1. 引言

附着在玻璃窗或者镜头上的雨滴会影响到背景图上某些部分的可见度，从而损失图像的信息。这些雨滴扮演着类似放大镜（图1）的作用，扩大了光线入射的视角，同时扭曲了原本入射相机光心的光线的光程；同时，在大多数情况下，相机则聚焦于背景面上，这就使得被雨滴遮挡的部分变得模糊。如何去除这些雨滴，从而提升图像的质量呢？目前已有许多方法致力于解决雨滴的检测和去除问题。方法 [14, 15, 8]旨在检测雨滴，却并不移除它们。

其他一些方法常用于检测和移除处理视频中的雨滴 [18, 20]，因此对于处理普通相机拍摄的单张带雨滴的图像并不适用。 Eigen等提出的方法 [1]尝试使用深度学习的方法，去除单张图像中的雨滴。但是它只能处理小的雨滴，并输出模糊的结果 [20]。 attentive GAN [12]能够处理大量出现的雨滴。通常来说，去雨滴问题是很棘手的，不仅因为这些被雨滴遮挡的区域的位置没有给定，这就需要确定雨滴的位置；而且这

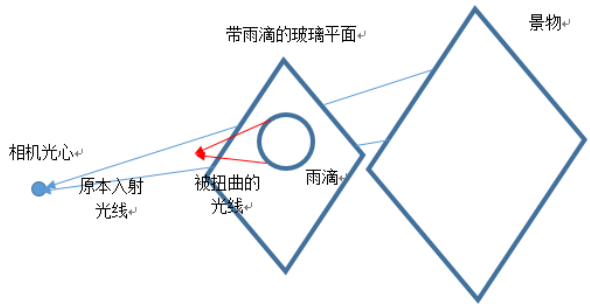

图 1. 雨滴造成的模糊

些被遮挡区域的背景信息几乎完全丢失了，重建是很困难的。当雨滴变大变密集时，这些问题就更难解决了。如果使用GAN网络来处理该问题，可将生成网络输出的去雨滴图像送入判别网络，判断其是否失真，并以此来衡量生成的去雨滴图像的质量。为了处理这个复杂的问题，attentive GAN首先生成了带雨滴图片对应的attention map。该注意力图是网络的核心，它将被用于引导生成网络聚焦于带雨滴的区域。注意力图是通过一个残差网络ResNet [5]，一个LSTM层 [17]和一些标准的卷积层生成的。

生成网络的第二部分是一个自动编码器，它接受带雨滴图像和注意力映射图作为输入，输出去雨滴后的图像。为了获取更广泛区域中的纹理信息，编码器的解码部分采用了多尺度损失。每个损失函数比较卷积层的输出和相应尺寸的真值图像的差异。除了该损失函数之外，我们还采用了感知损失来确保生成的图像和真值图像具有全局的高层特征的相似度。

在得到去雨滴的图像之后，判别网络将确认该图像是否失真。与一些图像修复的方法 [6, 9]类似，我们的判别网络在全局和局部两个层面来测算图像的失真

度。与图像修复不同之处在于，我们的问题是被雨滴遮挡的区域并未给定，因此判别网络无处发力。为解决该问题，attentive GAN复用了注意力映射图来引导判别网络做出判断。本文就注意力映射图是如何引导判别网络做出判断的进行了分析和修改，在ICIP 2019的去雨滴赛道test_a数据集上取得了一定的成果。

2. 相关工作

目前有许多工作聚焦于恶劣天气条件下的图像增强，这些工作主要解决的是由薄雾 [16, 4, 13]、雨线 [3, 2, 10, 19]的造成的图像清晰度损失问题。这些方法无法被直接用于去雨滴，因为附着在玻璃上的雨滴造成的模糊，和雨线、雾霭造成的模糊是不同的。前者是扭曲背景图像中某些部分造成的模糊，而后者则往往是对整幅图像进行了模糊处理。

Eigen等人提出的方法的基本思想是，使用带雨滴的和去雨滴的图像训练一个卷积神经网络。其CNN网络包含了3层，每层有512个神经元。虽然该方法对一些相对稀疏和细小的液滴污垢起作用，但是对于大而密的雨滴而言，它并不能产生很“干净”的结果。此外，输出的图像有时是有些模糊的。这些模糊可能是由于网络的参数量有限，以及在损失函数的设计上缺少足够的约束条件造成的。 attentive GAN采用了GAN作为网络骨干，这是因为GAN网络在图像复原方向近期取得了较好的成果 [7, 9]。与此方法类似，[7]在判别网络中采用了全局和局部分析。但与attentive GAN不同的是，在图像修复工作中，目标区域是给定的，因此局部分析很容易操作。另一个相似的结构是Pix2Pix，它将一张图片转换为另一张图片。它提出的conditional GAN不仅学习输入图像到输出图像的映射，而且学习用于训练映射的损失函数。该方法是一个全局的映射方法，并不特别适用于去雨滴问题。

A^2Net [11]网络(Adjacent Aggregation Network，邻接聚合网络)聚焦于由雨滴造成的模糊的机理。与attentive GAN网络中直接从RGB通道获取注意力映射图不同，A^2Net尝试着主要从YUV通道的Y通道（亮度通道）获取雨滴的位置。由于雨滴可以被视作一个在背景与相机感光元件之间额外放置的透镜，在被雨滴遮挡的位置，图像的亮度通道的值受到了显著的影响。据此观点，该网络更加注重Y通道。

如何印证这个观点呢？通过可视化200对真值图像与受雨滴影响图像质量下降的图像的残差像素值分布图2（在RGB通道和YUV通道分别可视化），该观点被印证是正确的。这样，原先在attentive GAN中通过一个复杂的RNN网络生成的attention map就可以通过我们的先验知识来更加快速地获取，而我们也可以进一步将原先这个图到图的复杂的映射问题，简化为分别处理Y通道和UV通道图像的问题。我们直接使用自编码器模型来处理这个问题。

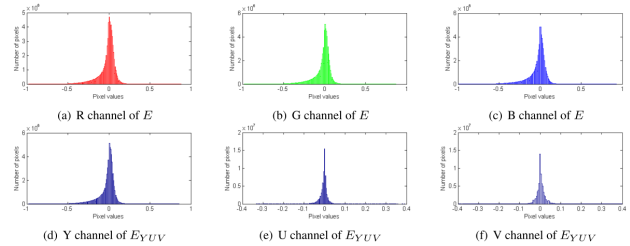

图 2. 可视化200对真值图像与受雨滴影响图像质量下降的图像的残差像素值分布

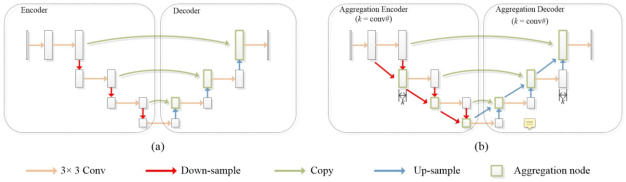

图 3. A^2Net中的skip connection

Loss	MSE		SSIM		Eq. (6)	
	SSIM	PSNR	SSIM	PSNR	SSIM	PSNR
testA	0.861	26.48	0.926	30.59	0.927	30.79
testB	0.785	24.07	0.834	25.33	0.834	25.50

图 4. A^2Net中的损失函数实验结果

原先可用于知识蒸馏的一些沙漏型模型，如UNet，可以在不同层级抽取特征；但是这些模型更擅长抽取高阶层的语义特征，而代价就是损失一部分浅层的细节特征，这与我们的诉求（希望网络提取和保存更多的细节特征）不符，因而直接将这些模型应用到我们的任务中效果并不是非常好。直接采用朴素的自编码器模型，并且添加类似于ResNet的skip connection机制图3，反而能收获奇效。

进一步探究损失函数的设计，我们发现MSE并不是一个足够好的选择。将SSIM与MSE相结合图4是一个不错的想法，并且在去雨滴任务中收到了很好的效

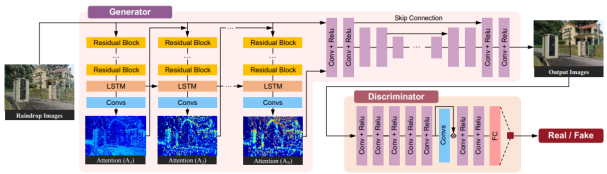

图 5. attentive GAN网络结构

果。其中损失函数设计成：

$$L_Y = \frac{1}{M}\sum_{i=1}^{M}(L_{MSE}(O_Y^i, T_Y^i)) + L_{SSIM}(O_Y^i, T_Y^i) \tag{1}$$

$$L_{UV} = \frac{1}{M}\sum_{i=1}^{M}(L_{MSE}(O_{UV}^i, T_{UV}^i)) + L_{SSIM}(O_{UV}^i, T_{UV}^i) \tag{2}$$

$$L = L_Y + \alpha \cdot L_{UV} \tag{3}$$

3. 算法介绍

带雨滴图像的构成. attentive GAN将带雨滴的图像建模为：

$$I = (1 - M) \odot B + R \tag{4}$$

其中，I表示输入图像，M表示一个二进制的掩膜矩阵。$M(x) = 1$表示像素x是雨滴的一部分，否则认为它是背景。B是背景图像，R是由雨滴造成的影响效果。操作符$\odot$表示按元素相乘。

基于式4，网络的目标是从输入I中获取背景图像B。因此我们需要知道M。我们定义$M(x) = 1$当且仅当带雨滴图像和无雨滴的真值图像的像素x位置处RGB三通道像素值之差的和超过30（见图8）。

生成-循环网络. 生成-循环网络是GAN网络（见图5）的第一部分，用于定位输入图像中需要施加注意力的区域。这些区域主要是有雨滴的和其周围的区域，关注它们将有助于编码器获取被遮挡区域的纹理信息。

生成网络的第一部分是生成-循环网络部分。该部分由N个同样的神经网络模块组成，每个模块从上至下分别由残差模块， LSTM模块和标准卷积层构成。该部分主要用于学习如何从一张带雨滴的图像中生成注意力映射图。我们将这个过程建模成一个多步操作，

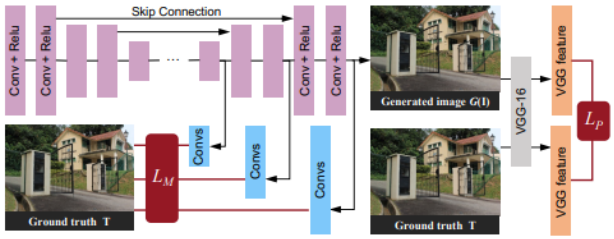

图 6. 上下文自编码器结构)。

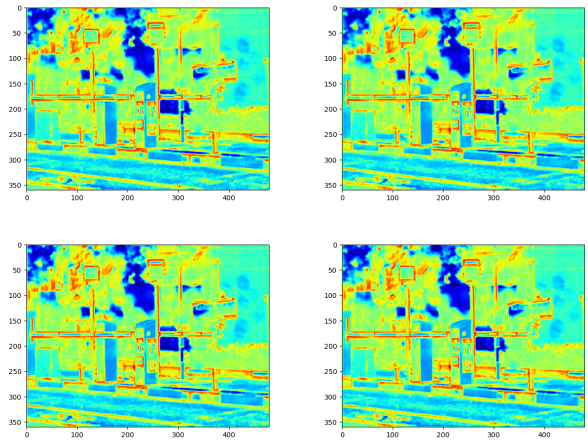

图 7. attentive map在循环网络中的4层输出结果，左上，右上，左下，右下分别表示第1-4层的输出

那么每一步都会对特定的一部分雨滴纹理施加注意力，而这些纹理内在是具有关联的，因此可以采用循环网络的结构去建模。每个模块的相应的损失函数也设计成利用前n层获取的注意力映射图的机制：

$$L_{ATT}(\{A\}, M) = \sum_{t=1}^{N} \theta^{N-t} L_{MSE}(A_t, M) \tag{5}$$

A_t表示第t个模块生成的注意力图。由于算力有限，这里采用N=4.

图7是在训练GAN网络时生成的注意力映射图。

上下文自编码器. 生成网络的第二部分是上下文自编码器（contextual autoencoder）图6。上下文自编码器的输入是待处理图像和生成-循环网络输出的最终注意力映射图。该深度自编码器具有16个卷积模块，以及为了防止输出模糊而设计的跳跃连接（skip connection）。

自编码器采用了多尺度损失函数和感知损失函数两个部分。前者从不同的解码层中提取不同尺度的输

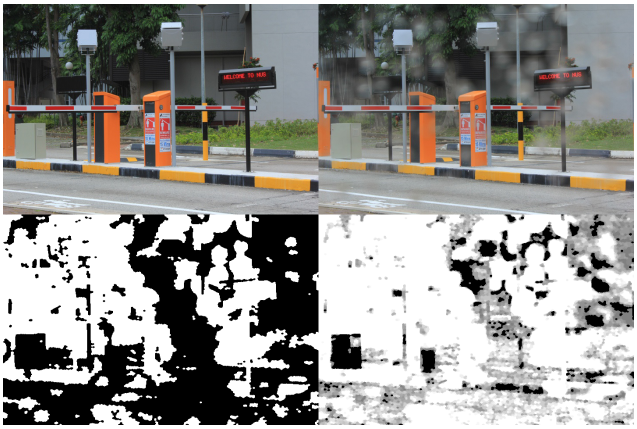

图 8. 由ground truth（左上），raindrop image（右上）生成的binary mask（左下）和smooth mask（右下）

出，与同尺度的真值图像做比较：

$$L_M(\{S\},\{T\}) = \sum_{i=1}^{M} \lambda_i L_{MSE}(S_i, T_i) \tag{6}$$

后者衡量了自编码器输出与真值图像在全局特征上的差别。该特征通过一个在 ImageNet上训练的VGG16网络来提取，损失函数形式为：

$$L_P(O,T) = L_{MSE}(VGG(O), VGG(T)) \tag{7}$$

这样就得到了生成网络部分的总损失函数：

$$\begin{aligned} L_G = 10^{-2} L_{GAN}(O) + L_{ATT}(\{A\}, M) \\ + L_M(\{S\},\{T\}) + L_P(O,T) \end{aligned} \tag{8}$$

其中GAN网络的损失为：

$$L_{GAN}(O) = log(1 - D(O)) \tag{9}$$

判别网络. 判别网络需要判定生成的图像是否失真，由于生成网络修改的主要部分是有雨滴的位置，因此如果判别网络也能定位到雨滴的位置，将有助于更加精确地判断图像的质量。attentive GAN在此复用了生成网络输出的最终的注意力映射图，使用它来引导判别网络施加更多注意力到相应的位置。

attentive GAN的判别器前几层网络是卷积层，用于获取生成的去雨滴图像中的浅层特征信息。后接一个卷积层，用于生成该图像对应的注意力映射图。然后将注意力图与浅层特征相乘，得到的施加了注意力的特征图作为输入，使得之后几层网络层便于判别该图像是否失真。因此，判别器的损失函数被设计成：

$$\begin{aligned} L_D(O,R,A_N) = -log(D(R)) - log(1 - D(O)) \\ + \gamma L_{map}(O,R,A_N) \end{aligned} \tag{10}$$

其中，L_{map}表示判别器网络前几层提取的注意力图和生成器生成的注意力图的损失函数：

$$\begin{aligned} L_{map}(O,R,A_N) = L_{MSE}(D_{map}(O), A_N) \\ + L_{MSE}(D_{map}(R), 0) \end{aligned} \tag{11}$$

这里的D_{map}表示判别器网络生成2D映射图的过程，γ设置为0.05。R是从真值图像中抽取出的一张图，0表示全零矩阵。式11表明如果输入的图像是带雨滴图，则判别器网络生成的注意力图在理想情况下是生成器网络生成的注意力图；如果输入图像是真值图像，则注意力图应该是全零矩阵。

我针对以上去雨滴的流程产生了几点思考：

1. 本系统架构的一个关键之处是表征雨滴位置的掩膜矩阵(mask)。原先训练数据集的掩膜矩阵是通过真值图像与带雨滴图像做差得到的，仅仅关注了带有雨滴的位置，而没有在雨滴的边缘处施加足够的注意力。我由此想到，能否计算出一个值更加平滑的掩膜矩阵，令带有雨滴的位置处矩阵值为1，在雨滴附近的位置值逐渐从1过渡到0（见图8）.

2. 在衡量生成的图像与真值图像的相似度的时候，perceptual loss能否很好地反应两者的全局范围的相似度呢？attentive GAN使用了一个在ImageNet上训练好的vgg16网络对整张图像进行特征提取，计算提取出的特征之间的相似度。我认为这不是一个好的方法，因为我们无法确定ImageNet 数据集中图像的分布与我们的带雨滴的图像之间是否具有相似性。

我参考了 [11]的思想：引入两张图像的结构相似性度量（Structural SIMilarity，SSIM）作为损失函数的一部分。因为SSIM的值域是[0, 1]，而且值越大表示两张图像越相似，因此我用1-SSIM作为损失函数的一部分加入式8：

$$\begin{aligned} L_G = 10^{-2} L_{GAN}(O) + L_{ATT}(\{A\}, M) \\ + L_M(\{S\},\{T\}) + L_{SSIM}(O,T) + L_P(O,T) \end{aligned} \tag{12}$$

其中：

$$L_{SSIM}(O,T) = 1 - SSIM(O,T) \tag{13}$$

	PSNR(test_a)	SSIM(test_a)	NIQE(test_a)	PSNR(test_b)	SSIM(test_b)	NIQE(test_b)
Torch版	31.516	0.9213	3.6448	24.9249	0.8091	4.1802
TF原版	25.7948	0.9043	3.7938	24.3363	0.8409	4.5349
TF-ssim-loss	26.1360	0.9109	3.6528	24.5049	0.8435	4.2612
TF-smooth-mask	26.3702	0.9151	3.5834	24.5303	0.8451	4.0724

表 1. 定量比较attentive GAN模型性能

4. 实验结果

实验步骤. 我在github上搜索，发现关于attentive GAN的论文的代码复现主要有Torch版本和Tensorflow版本这两个版本。由于Torch版本缺失了训练网络部分的代码，因此我采用了Tensorflow版本进行改进和实验。

平滑掩膜矩阵. 我在tf_io_pipline_tools.py文件的write_example_tfrecords()函数中，修改了_mask_image：

```
# original version
'''
  _mask_image[np.where(_diff_image >= 35)]
      = 1.
'''

# modified version
   _mask_image = _diff_image/30.0
   _mask_image[np.where(_diff_image > 30)]
       = 1.
   _mask_image[np.where(_diff_image < 15)]
       = 0.
```

考虑到显卡性能有限，我按照TensorFlow版代码的原版配置进行实验（配置在global_config.py中设定），总共训练了200000个epoch。

添加SSIM-loss. 我在attentive_gan_net.py中修改了computer_autoencoder_loss方法：

```
# original version
'''
loss = tf.add(lp_loss,lm_loss,name =
    'autoencoder_loss')
'''

# modified version
vgg_loss = tf.add(lp_loss,lm_loss,name =
    'vgg_loss')

# add ssim loss
ssim_max = tf.constant(1.0,tf.float32,name
    = "ssim_max")
ssim_val =
    tf.image.ssim(label_tensor,output_list[-1],max_val
    = 255)
ssim_loss =
    tf.subtrack(ssim_max,ssim_val[0],name =
    "ssim_loss")
loss = tf.add(vgg_loss,ssim_loss,name =
    "autoencoder_loss")
```

在此实验中，由于时间关系，我训练了145000个epoch。

我通过定量和定性比较来评测我的实验结果。

定量比较. 我在ICIP 2019 Challenge公布的test_a，test_b数据集上进行了实验，并与Tensorflow版本attentive GAN的作者训练的结果和Torch版本的结果进行了比较：

表1中的结果表明: 我的模型在两个数据集上的所有指标的性能都超过了原版Tensorflow的结果，这说明我的改动是有效果的，平滑掩膜矩阵和添加结构相似度损失确实起到了获取被雨滴遮挡部分的背景信息的作用。可惜的是，由于时间关系，我没来得及做完两个改动融合在一起的实验。

这里我的PSNR的指标一直比不上Torch版本的指标，我认为其中一个原因可能是TF版的图像经过了缩放，将720x480的图像缩放成480x360的图像，该步骤采用了opencv 中的插值方法，所以PSNR指标会有所下降。因此，我这里更多地是与TF版本作者训练的参数

图 9. 测试结果(左上为torch版本效果，右上为TF原版效果，左下为TF(smooth-mask)版效果，右下为TF(SSIM-LOSS)版效果)

的指标做对比。

定性比较. 定性比较的结果并不明显，见图9。

5. 总结

我认为attentive GAN的的工作是很有启发性的，它将GAN网络用于图像去雨滴的任务，映入了视觉领域经典的注意力机制，同时创造性地用循环网络结构来建模获取注意力映射图的过程。当站在这样的“巨人肩膀”上时，我们才能看得更长远，做出更有效果和价值的工作。

参考文献

[1] D. Eigen, D. Krishnan, and R. Fergus. Restoring an image taken through a window covered with dirt or rain. In *Proceedings of the IEEE International Conference on Computer Vision*, pages 633–640, 2013. 1

[2] 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*, 26(6):2944–2956, 2017. 2

[3] K. Garg and S. K. Nayar. Vision and rain. *International Journal of Computer Vision*, 75(1):3–27, 2007. 2

[4] K. He, J. Sun, and X. Tang. Single image haze removal using dark channel prior. *IEEE transactions on pattern analysis and machine intelligence*, 33(12):2341–2353, 2010. 2

[5] K. He, X. Zhang, S. Ren, and J. Sun. Deep residual learning for image recognition. In *Proceedings of the IEEE conference on computer vision and pattern recognition*, pages 770–778, 2016. 1

[6] S. Iizuka, E. Simo-Serra, and H. Ishikawa. Globally and locally consistent image completion. *ACM Transactions on Graphics (ToG)*, 36(4):107, 2017. 1

[7] P. Isola, J.-Y. Zhu, T. Zhou, and A. A. Efros. Image-to-image translation with conditional adversarial networks. In *Proceedings of the IEEE conference on computer vision and pattern recognition*, pages 1125–1134, 2017. 2

[8] H. Kurihata, T. Takahashi, I. Ide, Y. Mekada, H. Murase, Y. Tamatsu, and T. Miyahara. Rainy weather recognition from in-vehicle camera images for driver assistance. In *IEEE Proceedings. Intelligent Vehicles Symposium, 2005.*, pages 205–210. IEEE, 2005. 1

[9] Y. Li, S. Liu, J. Yang, and M.-H. Yang. Generative face completion. In *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition*, pages 3911–3919, 2017. 1, 2

[10] Y. Li, R. T. Tan, X. Guo, J. Lu, and M. Brown. Single image rain streak separation using layer priors. *IEEE transactions on image processing: a publication of the IEEE Signal Processing Society*, 2017. 2

[11] H. Lin, X. Fu, C. Jing, X. Ding, and Y. Huang. Aˆ 2net: Adjacent aggregation networks for image raindrop removal. *arXiv preprint arXiv:1811.09780*, 2018. 2, 4

[12] R. Qian, R. T. Tan, W. Yang, J. Su, and J. Liu. Attentive generative adversarial network for raindrop removal from a single image. In *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition*, pages 2482–2491, 2018. 1

[13] W. Ren, S. Liu, H. Zhang, J. Pan, X. Cao, and M.-H. Yang. Single image dehazing via multi-scale convolutional neural networks. In *European conference on computer vision*, pages 154–169. Springer, 2016. 2

[14] M. Roser and A. Geiger. Video-based raindrop detection for improved image registration. In *2009 IEEE 12th International Conference on Computer Vision Workshops, ICCV Workshops*, pages 570–577. IEEE, 2009. 1

[15] M. Roser, J. Kurz, and A. Geiger. Realistic modeling of water droplets for monocular adherent raindrop recognition using bezier curves. In *Asian Conference on Computer Vision*, pages 235–244. Springer, 2010. 1

[16] R. T. Tan. Visibility in bad weather from a single image. In *2008 IEEE Conference on Computer Vision and Pattern Recognition*, pages 1–8. IEEE, 2008. 2

[17] S. Xingjian, Z. Chen, H. Wang, D.-Y. Yeung, W.-K. Wong, and W.-c. Woo. Convolutional lstm network: A machine learning approach for precipitation nowcasting. In *Advances in neural information processing systems*, pages 802–810, 2015. 1

[18] A. Yamashita, I. Fukuchi, and T. Kaneko. Noises removal from image sequences acquired with moving camera by estimating camera motion from spatio-temporal information. In *2009 IEEE/RSJ International Conference on Intelligent Robots and Systems*, pages 3794–3801. IEEE, 2009. 1

[19] W. Yang, R. T. Tan, J. Feng, J. Liu, Z. Guo, and S. Yan. Deep joint rain detection and removal from a single image. In *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition*, pages 1357–1366, 2017. 2

[20] S. You, R. T. Tan, R. Kawakami, Y. Mukaigawa, and K. Ikeuchi. Adherent raindrop modeling, detectionand removal in video. *IEEE transactions on pattern analysis and machine intelligence*, 38(9):1721–1733, 2015. 1