Improving Robustness of Semantic Segmentation to Motion-Blur using Class-Centric Augmentation
SUPPLEMENTARY MATERIAL

Aakanksha
Indian Institute of Technology Madras
aakankshajha30@gmail.com

A. N. Rajagopalan
Indian Institute of Technology Madras
raju@ee.iitm.ac.in

In this supplementary material, we include the following.

1. A discussion on the choice of blur kernels - Linear or Non-Linear?
2. An ablation for the choice of 'p', the probability of blurring an image during training.
3. A class-wise comparison of quantitative performance gains of our approach over baseline methods on PASCAL VOC [4] and Cityscapes [3].
4. Additional qualitative results on synthetic space-invariant blur for PASCAL VOC [4] and Cityscapes [3] datasets.
5. Additional qualitative results on real blur for GOPRO [7] and REDS [6] datasets.

All the tables, figures and sections in this supplementary are numbered starting with an 'S' to make clear distinctions between the contents in the main paper and supplementary. The best results are highlighted in bold in all the tables.

S1. Linear v/s Non-Linear Blur Kernels

In our augmentation strategy, we synthesize space-variant class-centric blur as well as space-invariant synthetic motion blur. Since class-centric blurring is meant to model dynamic scene blur, using linear blur may seem more appropriate. We argue that since our work attempts to model both camera motion blur (which is typically non-linear [1]) and dynamic scene blur (which is pre-dominantly linear [5]) using a single augmentation strategy, it becomes imperative to include both linear and non-linear blur kernels during training. The blur kernel generation method detailed in Sec.3.1.1 in the main paper refers to 3 anxiety levels which control the non-linearity of the kernels generated, with lower anxiety level corresponding to less non-linearity. The anxiety level of $a = 0.00005$ used while generating blur kernels, corresponds to an approximately linear blur which makes our set inclusive of both non-linear and linear blur kernels. Additionally, for non-rigid objects like humans, where motion blur can be caused by fast movement of body parts, using non-linear kernels is better.

To establish the effectiveness of our augmentation strategy even if linear kernels are used to model blur, we perform an ablation study where we train DeepLabv3+ [2] with MobileNetv2 [8] backbone on PASCAL VOC dataset using only linear blur kernels for different blur levels. We generated blur kernels with 3 levels of exposure as detailed in the paper but restricting anxiety to the lowest value of $a = 0.00005$. All the training setup remains the same and the model is trained with $p = 0.5$, the probability of an image being blurred. The performance metric used is standard mIoU as used in the main paper. The results are shown in Table S1. It can be clearly seen that modeling blur using linear blur kernels gives similar performance as non-linear blur kernels for clean images and blur level L1. A slight performance drop is observed at higher blur levels L2 and L3 when using linear blur kernels.

Table S1. Quantitative comparisons for training using CCMBA with linear vs non-linear blur kernels.

Blur Kernel Type	Clean	L1	L2	L3
Linear	**69.6**	**68.5**	65.9	60.5
Non-Linear	69.3	68.2	**66.6**	**61.5**

S2. Ablation for the blurring probability hyper-parameter p

In this section, we document results for blurring with different probabilities during training. We chose $p = 0.5$ during the training of all our models in the main paper.

Table S2. Quantitative comparisons for training with CCMBA using different p values.

Probability of Blurring	Clean	L1	L2	L3
$p = 0.3$	**69.6**	67.9	64.9	60.2
$p = 0.5$	69.3	**68.2**	**66.6**	61.5
$p = 0.7$	68.7	**68.2**	65.2	**62.0**
$p = 0.9$	67.7	67.7	65.4	61.6

Table S3. Class-wise comparison of mIoUs with baselines on PASCAL VOC for clean images.

	Background	Aeroplane	Bicycle	Bird	Boat	Bottle	Bus	Car	Cat	Chair	Cow	Dining Table	Dog	Horse	Motorbike	Person	Potted Plant	Sheep	Sofa	Train	Tv Monitor
No-Retraining	93.26	79.71	41.15	85.6	**71.76**	72.56	90.81	83.25	**91.19**	32.57	83.73	**60.87**	**86.43**	82	82.35	82.01	57.88	**83.69**	49	84.52	68.04
Finetune	88.12	80.01	34.14	70.97	46.73	40.81	79.58	76.4	74.7	18.56	59.63	38.34	67.33	65.52	65.52	75.18	34.52	66.23	34.93	60.03	51.92
MBA	92.39	81.98	40.46	**86.28**	48.81	65.38	91.38	85.34	90.19	35.53	78.96	58.76	84.7	83.55	79.77	83.48	46.43	76.19	**49.57**	80.78	70.89
Ours	**94.01**	**86.4**	**41.46**	86.13	62.26	**73.22**	**92.64**	**87.18**	89.1	**36.54**	**85.51**	59.54	83.08	**84.35**	**82.68**	**85.7**	**62.28**	80.67	43.97	**87.21**	**77.03**

Table S4. Class-wise comparison of mIoUs with baselines on PASCAL VOC for images with L1 blur.

	Background	Aeroplane	Bicycle	Bird	Boat	Bottle	Bus	Car	Cat	Chair	Cow	Dining Table	Dog	Horse	Motorbike	Person	Potted Plant	Sheep	Sofa	Train	Tv Monitor
No-Retraining	91.91	74.52	37.23	75.02	61.22	69.85	88.18	81.68	85.09	29.13	73.32	51.12	80.17	73.77	79.61	78.3	49.65	71.98	46.71	79.31	67.57
Finetune	91.42	78.85	39.31	82.53	56.09	57.6	90.32	83.93	86.64	33.53	69.68	54.89	82.82	76.52	78.97	79.62	42.9	75	48.67	79.81	62.87
MBA	91.97	80.87	39.2	82.81	48.27	65.55	**92.15**	83.89	**88.74**	32.28	74.25	**59.07**	**83.25**	80.08	77.66	81.47	44.5	72.89	**49.55**	81.66	66.92
Ours	**93.58**	**83.23**	**40.82**	**84.25**	**63.37**	**76.22**	90.97	**85.1**	88.02	**38.72**	**83.89**	57.31	82.73	**82.6**	**81.26**	**83.43**	**61.9**	**79.03**	42.13	**85.9**	**77.7**

Table S5. Class-wise comparison of mIoUs with baselines on PASCAL VOC for images with L2 blur.

	Background	Aeroplane	Bicycle	Bird	Boat	Bottle	Bus	Car	Cat	Chair	Cow	Dining Table	Dog	Horse	Motorbike	Person	Potted Plant	Sheep	Sofa	Train	Tv Monitor
No-Retraining	89.38	68.48	31.74	64.3	42.77	63.67	77.97	72.07	74.07	24.84	61.92	36.82	67.74	65.17	66.59	72.44	42.8	46.19	37.76	67.8	60.93
Finetune	91.03	76.74	37.44	78.05	55.61	58.95	89.08	81.41	85.12	32.51	65.01	53.5	**81.42**	70.88	76.13	78.26	42.07	73.74	**47.1**	76.85	62
MBA	91.58	80.03	37.02	79.58	47.86	63.58	89.12	82.05	85.65	32.17	71.47	56.37	78.39	75.02	76.31	79.69	48.04	70.69	46.5	77.93	66.94
Ours	**93.07**	**80.28**	**39.17**	**82.18**	**58.92**	**71.86**	**89.88**	**82.73**	**86.27**	**38.04**	**79.01**	**57.11**	79.42	**79.33**	**79.46**	**81.87**	**58.25**	**75.89**	40.82	**84.44**	**76.25**

Table S6. Class-wise comparison of mIoUs with baselines on PASCAL VOC for images with L3 blur.

	Background	Aeroplane	Bicycle	Bird	Boat	Bottle	Bus	Car	Cat	Chair	Cow	Dining Table	Dog	Horse	Motorbike	Person	Potted Plant	Sheep	Sofa	Train	Tv Monitor
No-Retraining	85.41	61.53	28.05	51.21	24.41	50.75	60.53	57.82	57	16.62	28.85	14.08	52.71	41.45	41.02	60.22	30.26	29.42	25.22	41.03	51.76
Finetune	90.08	69.99	34.81	74.15	52.83	57.73	82.83	77.07	81.05	28.99	63.99	50.53	76.66	66.42	71.01	74.71	43.67	**74.27**	43.53	70.99	62.19
MBA	90.52	72.91	34.19	76.44	46.1	59.09	82.37	77.55	82.39	27.8	69.52	54.03	**77.16**	71.25	69.62	75.72	45.61	68.57	**43.95**	72.47	67.45
Ours	**92.16**	**77.34**	**35.88**	**78.18**	**54.74**	**65.68**	**88.04**	**79.7**	**82.61**	**34.34**	**77.62**	**54.05**	73.69	**75.67**	**73.14**	**79.05**	**55.95**	73.5	38.96	**82.91**	**69.02**

Table S7. Class-wise comparison of mIoUs with baselines on Cityscapes for clean images.

	Road	Sidewalk	Building	Wall	Fence	Pole	Tr. Light	Tr. Sign	Vegetation	Terrain	Sky	Person	Rider	Car	Truck	Bus	Train	Motorcycle	Bicycle
No Retraining	97.93	**83.77**	91.82	46.47	**59.01**	61.95	66.72	76.48	92.32	63.84	94.57	80.60	59.99	94.39	**74.78**	**85.85**	**71.80**	59.36	75.72
Finetuning	97.25	79.42	90.45	32.62	49.39	59.07	58.88	74.12	90.46	56.40	93.08	78.44	56.99	93.27	69.40	78.50	62.60	48.83	72.04
MBA	97.86	83.29	92.00	41.98	53.98	64.06	68.37	76.44	92.01	63.02	94.82	81.21	61.90	94.27	70.14	82.57	66.01	60.68	75.40
Ours	**97.88**	83.60	**92.28**	**47.73**	56.78	**65.54**	**70.18**	**79.10**	**92.35**	**64.48**	**94.89**	**82.05**	**62.30**	**94.65**	72.07	85.11	66.34	**63.72**	**76.96**

Table S8. Class-wise comparison of mIoUs with baselines on Cityscapes for images with L1 blur.

	Road	Sidewalk	Building	Wall	Fence	Pole	Tr. Light	Tr. Sign	Vegetation	Terrain	Sky	Person	Rider	Car	Truck	Bus	Train	Motorcycle	Bicycle
No Retraining	97.39	80.71	90.00	40.19	52.16	56.55	59.99	71.32	90.40	59.97	91.56	75.87	52.28	93.20	68.27	78.62	55.30	53.55	70.82
Finetuning	97.74	82.31	91.51	46.46	53.89	59.65	65.40	74.98	91.74	61.63	94.14	78.91	57.76	93.95	72.39	83.65	**72.44**	58.10	73.77
MBA	97.75	82.39	91.43	41.49	51.28	61.27	66.90	75.08	91.63	62.27	94.41	79.57	60.32	94.00	73.38	81.32	64.99	57.84	73.59
Ours	**97.76**	**82.67**	**91.87**	**48.61**	**54.31**	**63.21**	**68.99**	**77.54**	**92.03**	**63.94**	**94.50**	**80.68**	**61.08**	**94.33**	**74.38**	**84.72**	65.92	**64.24**	**75.14**

Table S9. Class-wise comparison of mIoUs with baselines on Cityscapes for images with L2 blur.

	Road	Sidewalk	Building	Wall	Fence	Pole	Tr. Light	Tr. Sign	Vegetation	Terrain	Sky	Person	Rider	Car	Truck	Bus	Train	Motorcycle	Bicycle
No Retraining	95.26	71.99	84.01	19.99	34.06	46.92	45.14	59.59	83.91	42.93	86.46	64.94	35.60	89.45	54.48	65.28	25.67	40.05	57.96
Finetuning	97.58	81.02	90.84	45.21	50.65	56.68	62.47	72.12	91.25	60.29	93.71	77.36	55.92	93.49	69.61	80.09	**64.94**	56.07	71.70
MBA	**97.62**	**81.40**	90.81	39.13	48.83	58.55	63.79	72.60	91.07	61.37	94.21	77.74	**58.78**	93.51	71.40	78.69	61.37	54.23	71.33
Ours	97.58	81.24	**91.32**	**46.03**	**51.23**	**60.55**	**66.19**	**75.29**	**91.59**	**62.69**	**94.22**	**78.52**	58.66	**93.92**	**73.89**	**81.55**	60.04	**61.16**	**73.09**

Table S10. Class-wise comparison of mIoUs with baselines on Cityscapes for images with L3 blur.

	Road	Sidewalk	Building	Wall	Fence	Pole	Tr. Light	Tr. Sign	Vegetation	Terrain	Sky	Person	Rider	Car	Truck	Bus	Train	Motorcycle	Bicycle
No Retraining	86.87	56.01	70.76	4.38	12.99	31.88	27.79	45.03	71.42	26.42	74.32	48.92	18.21	77.97	18.91	49.24	15.42	15.86	34.33
Finetuning	97.16	78.11	89.58	41.96	43.82	50.35	57.56	66.99	90.12	59.36	92.96	72.96	50.88	91.95	67.53	75.50	53.60	50.67	67.75
MBA	97.24	78.69	89.76	37.83	45.17	53.13	58.71	67.68	90.09	59.76	93.38	74.02	54.30	92.14	65.79	73.90	**57.55**	46.69	67.09
Ours	**97.31**	**79.27**	**90.29**	**43.90**	**47.89**	**54.58**	**60.85**	**70.71**	**90.62**	**61.11**	**93.56**	**75.12**	**54.58**	**92.67**	**68.89**	**78.15**	55.05	**54.34**	**69.02**

As can be seen from Table S2, as we increase the probability of an image being class-centric blurred, the performance on clean images sees a decrease but performance on higher levels of blur increases. So, $p = 0.5$ is a good choice during training.

S3. Class-wise evaluation of performance for space-invariant blur

In this section, we show class-wise quantitative comparisons of our method with baseline methods on PASCAL VOC (Table S3 - S6) and Cityscapes (Table S7 - S10) datasets to demonstrate that performance gains are achieved across most classes using our approach. In Table S3 and Table S7, we compare the performance of all baselines and our approach for clean sharp images. In Table S3, we can see that our method performs slightly worse than the 'No Retraining' baseline on the classes - Boat, Cat, Dining Table, Dog, Sheep and Sofa while improved performance scores are observed for all other classes for PASCAL VOC dataset. Similarly, in Table S7 for the Cityscapes dataset, our method performs slightly worse than the 'No Retraining' baseline on the classes - Sidewalk, Fence, Truck, Bus, Train while improved performance scores are observed for all other classes.

In Table S4 and Table S8, we provide comparisons for images with L1 level of blur for PASCAL VOC and Cityscapes datasets respectively. For certain classes, like Bus, Cat, Dining Table, Dog and Sofa, in PASCAL VOC, the 'MBA' baseline seems to give the best performance, while our method performs best for all the remaining

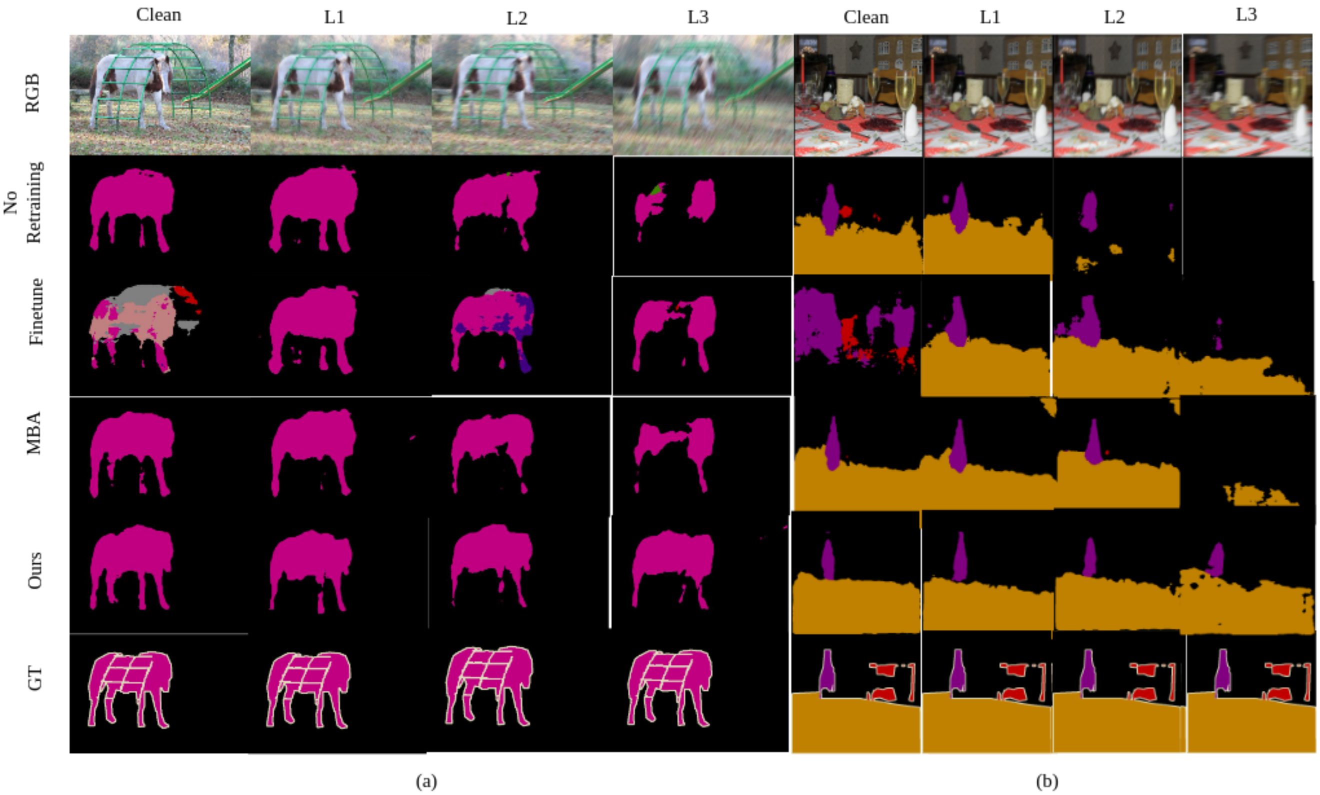

Figure S1. Qualitative results for space-invariant motion blur for DeepLabv3+ on PASCAL VOC. Note that our method consistently outperforms all baselines.

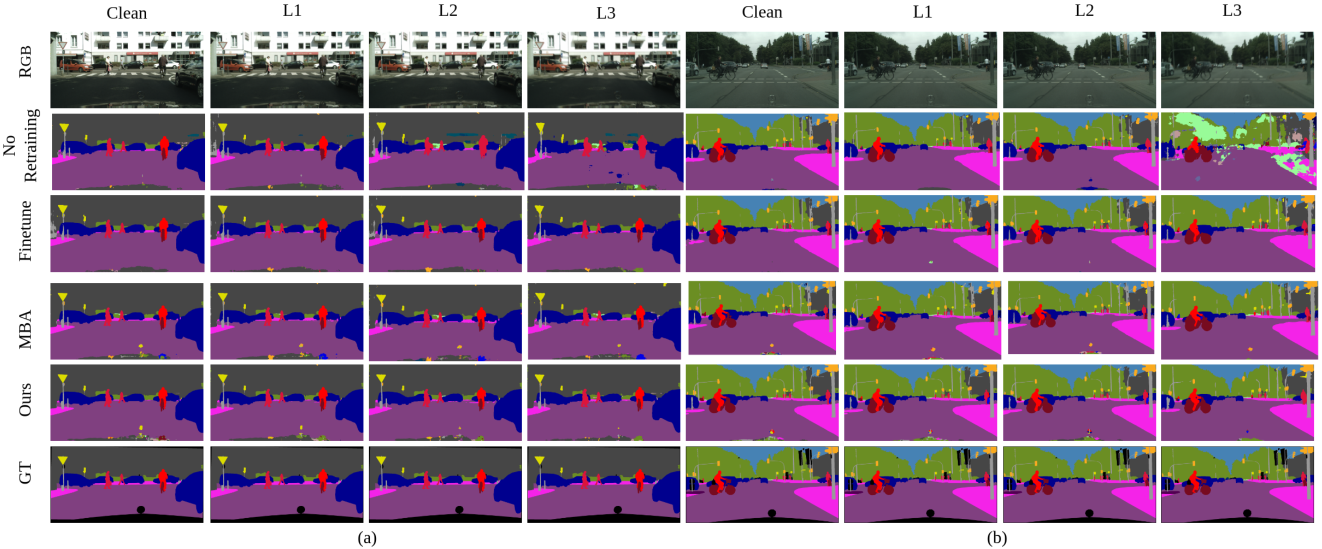

Figure S2. Qualitative results for space-invariant motion blur for DeepLabv3+ on Cityscapes.

classes. On the other hand, for Cityscapes, our model outperforms all baselines on all classes.

In Table S5, we compare the performances for images with L2 level of blur for PASCAL VOC dataset and our method performs better than all baselines for all classes except, Dog and Sofa, where the 'Finetune' baseline performs best. For Cityscapes, we compare the performances on images with L2 level of blur in Table S5 and our method outperforms all baselines on all classes except Road, Sidewalk, Rider and Train where our approach lags only by a small margin.

Lastly, in Table S6 and Table S10, we compare the performances for images from PASCAL VOC and Cityscapes with L3 level of blur. Our method outperforms all base-

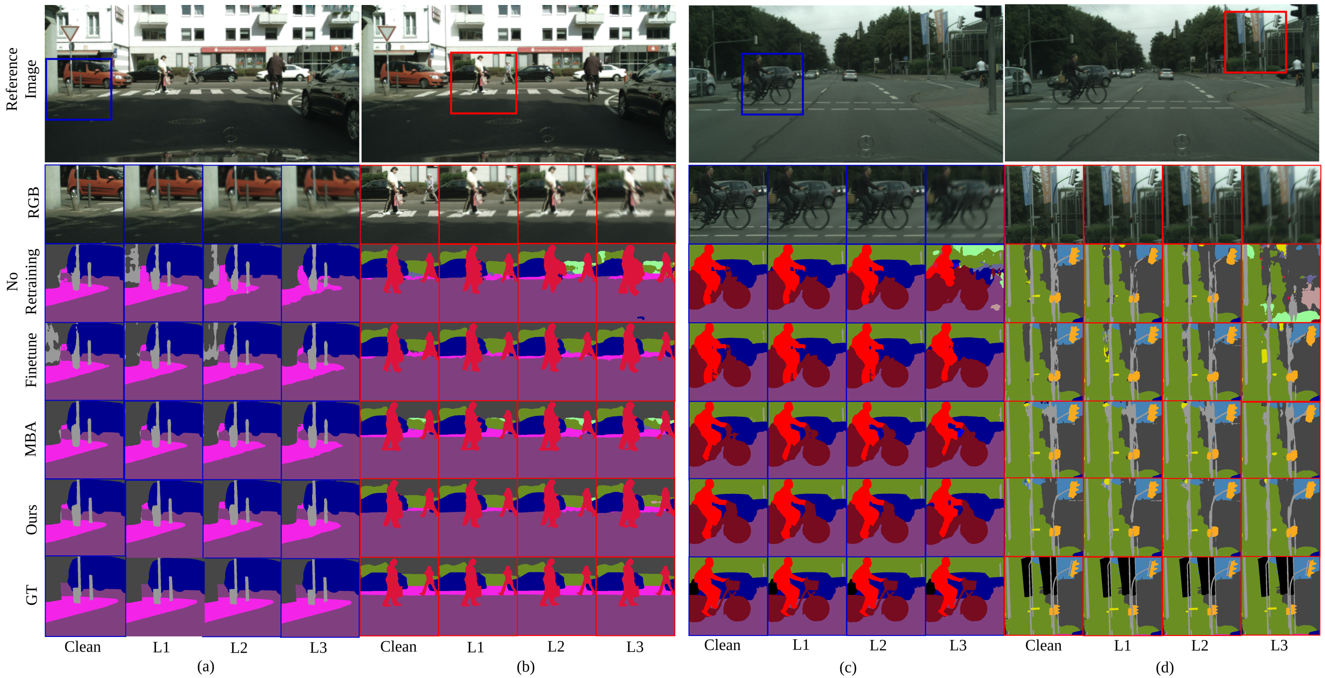

Figure S3. Zoomed in regions from Fig. S2. Reference image is used only to depict the cropped region on the sharp image. Two regions are taken for each image and are highlighted by a red square and a blue square respectively. Note that our method captures finer details better and is more consistent across sharp and blurred images when compared to baseline methods.

lines for all classes in PASCAL VOC except, Dog and Sofa, where the 'Finetune' and 'MBA' baselines perform best respectively. On Cityscapes, our method outperforms all baselines for all classes except Train where the 'MBA' baseline performs best.

So, our method improves performance for almost all classes in the presence of different levels of blur when compared to baseline methods.

S4. Qualitative Results for Synthetic Blurred Images

In this section, we provide additional results on PASCAL VOC and Cityscapes datasets. For Cityscapes, we show zoomed in cropped regions to highlight the smaller regions because of the large image size.

Fig. S1(a) and Fig. S1(b) are both images taken from PASCAL VOC datset. Our method performs better than all the baselines, especially, at blur level L3 and the performance drop is very small as we move from clean images to blur level L3 for our method.

In Fig. S2, we show results for Cityscapes dataset for two images. Due to the large size of the images, the degradations due to blur are not very evident in baselines other than 'No Retraining'. For better visualization, we crop 2 square regions for each of the images and show the zoomed in results in Fig. S3 where (a) and (b) are crops of Fig. S2(a), and, (c) and (d) are crops of Fig. S2(b). On careful observation, the first thing to notice is the consistency of our results across clean and blur images as opposed to other baselines for each of the crops. For S3(a), our results are significantly better than 'Finetune' baseline but comparable to 'MBA' baseline. But on considering, S3(b), our results are better than 'MBA' baseline and comparable to 'Finetune' baseline. So for the same image, different baseline methods give better performance in different regions but our method performs consistently well throughout.

Now considering Fig. S3(c), we draw attention to the handle of the cycle which is more consistently picked up by our approach across all blur levels. In Fig. S3(d), all the baseline models confuse parts of the background with the thin lamp-post which is better segmented using our method, especially, for blur level L3.

S5. Qualitative Results on GOPRO and REDS for Real Blur

We show additional results for GOPRO and REDS dataset using DeepLabv3+ model trained with our approach on PASCAL VOC dataset. Fig. S4(a) and (b) are examples from GOPRO and (c) is an example from REDS. In Fig. S4(a), we can clearly see that good segmentation maps are obtained for the sharp image using all the methods but for the blurred counterpart, the baseline models struggle. Our approach gives the best segmentation map for the blurred image with finer details like the legs of the person evident

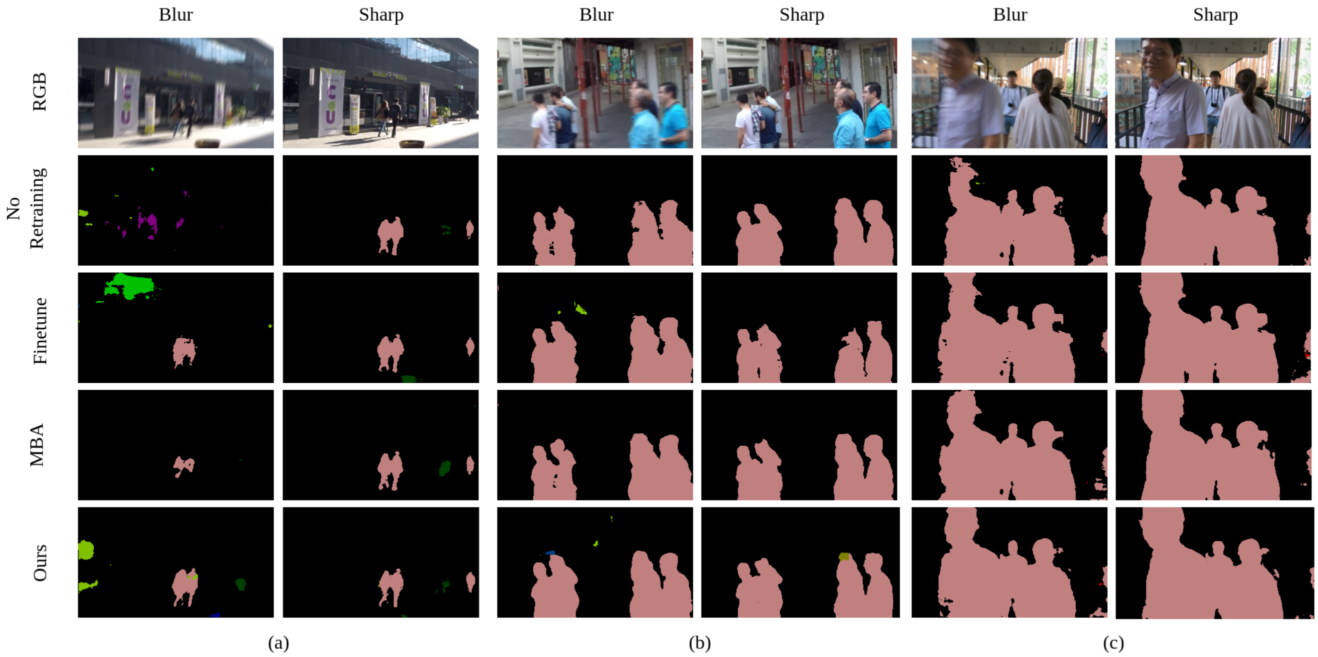

Figure S4. Qualitative results for real motion blur for DeepLabv3+ on GOPRO and REDS.

from the map itself. In Fig. S4(b), the 'No Retraining' baseline performs best on sharp image while 'Finetune' baseline gives best results on blurred image. Our method gives comparable results to 'Finetune' baseline for blurred image and 'No Retraining' baseline for corresponding sharp image. In Fig. S4(c), note the heavily blurred man on the left. While other baseline methods struggle to segment out this region, owing to the spatial nature of the blur, our approach does a good job of segmenting out this man, while giving comparable performance to the baselines on the sharp image. These results clearly show that our method performs better than all the baselines on these real world blur images while also being able to retain performance on sharp images.

References

[1] Giacomo Boracchi and Alessandro Foi. Modeling the performance of image restoration from motion blur. *IEEE Transactions on Image Processing*, 21(8):3502–3517, 2012. 1

[2] Liang-Chieh Chen, Yukun Zhu, George Papandreou, Florian Schroff, and Hartwig Adam. Encoder-decoder with atrous separable convolution for semantic image segmentation. In *Proceedings of the European conference on computer vision (ECCV)*, pages 801–818, 2018. 1

[3] Marius Cordts, Mohamed Omran, Sebastian Ramos, Timo Rehfeld, Markus Enzweiler, Rodrigo Benenson, Uwe Franke, Stefan Roth, and Bernt Schiele. The cityscapes dataset for semantic urban scene understanding. In *Proc. of the IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*, 2016. 1

[4] M. Everingham, L. Van Gool, C. K. I. Williams, J. Winn, and A. Zisserman. The PASCAL Visual Object Classes Challenge 2007 (VOC2007) Results. http://www.pascal-network.org/challenges/VOC/voc2007/workshop/index.html, 2007. 1

[5] Peidong Liu, Joel Janai, Marc Pollefeys, Torsten Sattler, and Andreas Geiger. Self-supervised linear motion deblurring. *IEEE Robotics and Automation Letters*, 5(2):2475–2482, 2020. 1

[6] Seungjun Nah, Sungyong Baik, Seokil Hong, Gyeongsik Moon, Sanghyun Son, Radu Timofte, and Kyoung Mu Lee. Ntire 2019 challenge on video deblurring and super-resolution: Dataset and study. In *CVPR Workshops*, June 2019. 1

[7] Seungjun Nah, Tae Hyun Kim, and Kyoung Mu Lee. Deep multi-scale convolutional neural network for dynamic scene deblurring. In *CVPR*, July 2017. 1

[8] Mark Sandler, Andrew Howard, Menglong Zhu, Andrey Zhmoginov, and Liang-Chieh Chen. Mobilenetv2: Inverted residuals and linear bottlenecks. In *Proceedings of the IEEE conference on computer vision and pattern recognition*, pages 4510–4520, 2018. 1