Bringing Alive Blurred Moments (Supplementary material)

Kuldeep Purohit[1] Anshul Shah[2*] A. N. Rajagopalan[1]
[1] Indian Institute of Technology Madras, India [2] University of Maryland, College Park
kuldeeppurohit3@gmail.com, anshulb@cs.umd.edu, raju@ee.iitm.ac.in

In this document, we further analyze our video extraction and deblurring networks followed by additional qualitative results and comparisons for video generation and single image deblurring. *Note that videos can be viewed by clicking on the image, when document is opened in Adobe Reader.*

Video Results corresponding to the Figs. 6, 7, and 8 of our main paper are provided in Figs. S1, S2, and S3, respectively.

S1. Ablation Studies for Video Extractor

This section describes the experiments that lead to our design choices and involves quantitative comparisons of different configurations for training our network. In the following subsections, reported scores are calculated over 250 test examples. Each test sequence is composed of 7 frames of resolution 1280×704, taken from the 11 test videos from the GoPro Dataset [23] (the same test set was used for video extraction evaluation in section 3.3 of the main paper).

S1.1. Effect of different losses

We experimented with the effect of various losses while training the RVE-RVD pair for video reconstruction, and analyzed their performance based on the average reconstruction error values. In the first configuration, both frame reconstruction loss and the TV loss were enforced at only a single scale (corresponding to $j = 4$ in Eqns. (4) and (5) of our main paper). This model did not perform well on sequences containing complex motion, and resulted in local fluctuations in motion, leading to a total MSE of 0.595. In the second configuration, the reconstruction losses were present at all the 4 scales but no cost was enforced (TV loss) on the predicted flows. On visual inspection, we found that it resulted in flows containing 'noise' (reduced spatial smoothness) thus resulting in slightly higher reconstruction errors (MSE 0.568). When we included multi-scale frame reconstruction and TV loss with relative weights as mentioned in section 2.4 (which is the final loss setting that we use), it led to a total MSE of 0.542.

*Work done while at Indian Institute of Technology Madras, India.

S1.2. Non-recurrent CNN

For the task of video reconstruction, we also experimented by replacing our RVE with a 3D-convolutional network. Specifically, we chose an architecture with four $3 \times 3 \times 3$ convolutional layers and 3D-maxpool operations to finally yield a feature of same dimension as the motion embedding returned by RVE. The number of filters was chosen to approximately match the total number of parameters in our RVE. The 7 frames were stacked along the temporal dimension and fed to this CNN-RVD pair, which is trained for the video reconstruction task. However, it was not quite successful in reproducing local motion in dynamic videos and led to a higher MSE of 0.595 against 0.542 achieved using RVE. These observations reaffirm the effectiveness of recurrent modules for our task.

S1.3. BIE sans Sharp Image

We trained a version of our network wherein the BIE was not fed a sharp image. This network led to higher average error 48.75 (using the loss function described in Eqn. (6)) than our main network. We suspect that this is because the availability of sharp intensities provides a better reference for measuring the blur at each pixel. In other words, the ill-posed-ness of the blur prediction task reduces with the availability of the sharp image.

S1.4. Direct BIE-RVD Training

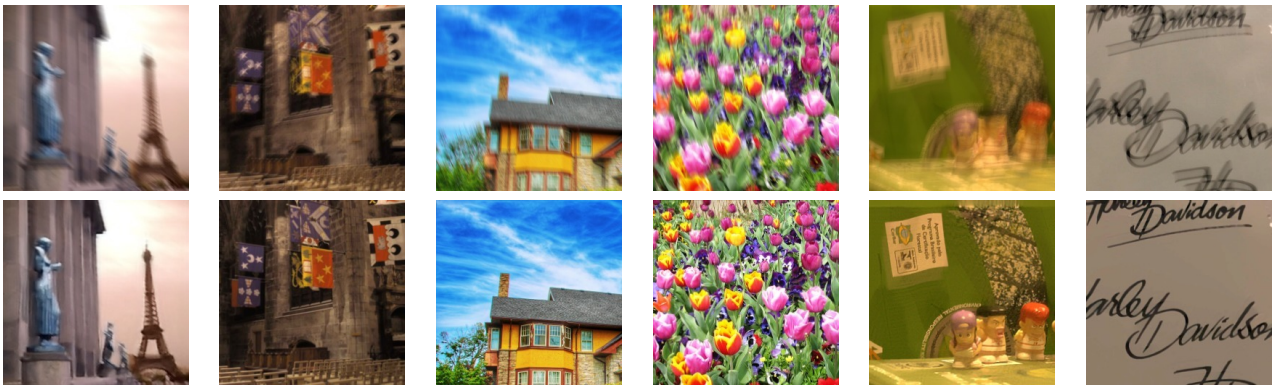
We found that direct training of BIE and RVD from scratch poses a formidable challenge and the performance is below par (average error 50.08). The improvement due to pre-training of RVD is attributed to the fact that video reconstruction task does not suffer from ambiguity and hence RVD can be trained optimally using the conventional loss (eq.5 in the main paper). Moreover, our approach of training RVD for the surrogate task of extracting motion representation of short video sequences has the advantage of rendering the learned motion representation interpretable.

S1.5. Direct Intensity Estimation

We investigate the advantage of motion-flow learning, instead of direct intensity estimation. In this experiment,

(a) (b) (c) (d) (e) (f)

Figure S1. Comparisons of our video extraction results with [13] on motion blurred images obtained from the test dataset of [23]. The first row shows the blurred images while the second and third rows show videos generated by our method and [13], respectively. *Videos can be viewed by clicking on the images, when document is opened in Adobe Reader.*



(a) (b) (c) (d) (e) (f)

Figure S2. Video generation from images blurred with global camera motion from datasets of [7,15] and [19]. First row shows the blurred images. The generated videos using our method are shown in second row.

the modification involved removal of the transformer layers (described in section 2.2 of main paper) from RVD, such that the modified network (referred to as $\text{RVD}_{\text{direct}}$) directly estimates pixel intensities (instead of motion-flows). Our experiments revealed an issue with this approach: such an RVE-$\text{RVD}_{\text{direct}}$ pair learns an identity mapping (i.e. the sharp intensities are directly propagated from RVE to $\text{RVD}_{\text{direct}}$ through the hidden features). Such an autoencoder does not achieve the goal of learning motion representations and hence fails to be useful in guided training of BIE for the video extraction task. To avoid this limitation, we directly trained the BIE-$\text{RVD}_{\text{direct}}$ pair to estimate the intensities of the sharp video from a blurred image. Results of this network are shown in Fig. S4, where it can be observed that regressing to the intensities of each frame leads to distortions in the image content and unpleasant artifacts. The method of [13] attempts to address this issue with the help of more complex losses (including perceptual loss and adversarial loss), but still fails in cases of large motion, as demonstrated in section 3.3 of the main paper. Note that our original approach (which performs motion-flow estimation) explicitly enforces motion relevant feature learning by predicting pixel-level motion-flows instead of the intensities. The encoding learnt by our RVE-RVD cannot be an identity mapping since robust optical flow prediction is not possible without capturing the dynamics of the scene

S2. Analysis of Our Deblurring Network

S2.1. Implementation Details

In Table S1, we provide layer-wise details of parameters involved in our deblurring architecture. Symbols H and W represent the height and width of the input blurred image.

S2.2. Effect of growth-rate (GR)

Growth-rate of the densely connected encoder layers is the key hyper-parameter of our deblurring network. To find its optimal value, we design and train 3 versions of the network with different growth-rates. Fig. S5 shows comparisons of the convergence process of these 3 models. It can

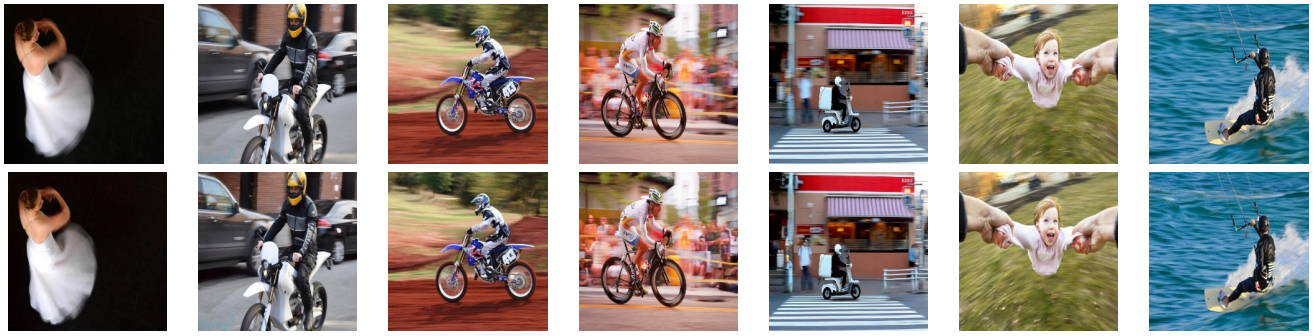


Figure S3. Video generation results on real motion blurred images from dataset of [32]. The first row shows the blurred images. Second row contains the extracted videos with our method.

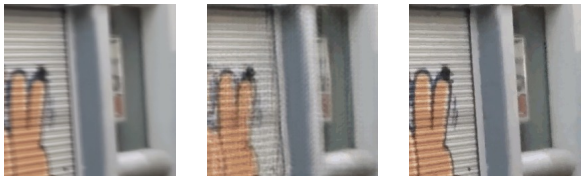


Figure S4. A blurred image (left) from the dataset of [23] and the corresponding videos obtained using a baseline (middle) which estimates the intensities of each frame and ground-truth video (right).

be observed that the training performance get better with increase in growth-rate. We chose GR=32 in our proposed model, since the improvement beyond 32 is marginal and it serves as a good balance between efficiency and performance.

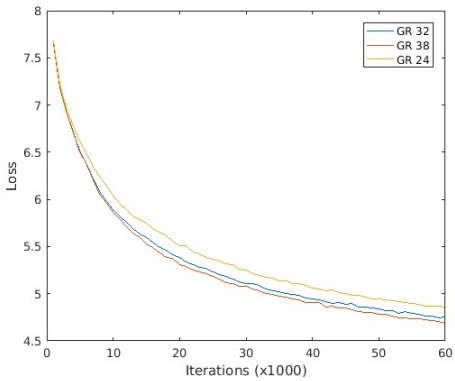


Figure S5. Training loss curves corresponding to models with different growth-rates.

S2.3. Effect of local residual connections

In Fig. S6, we compare the training performance of our deblurring network with a version of it that does not contain local residual connections [44]. These connections exist between the input and output of each dense block in the encoder and contribute to the flow of information and gradient. It can be inferred that inclusion of such connections leads to lower errors, validating that this component efficiently improves the performance.

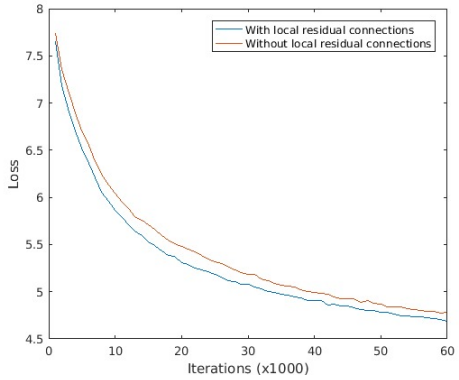


Figure S6. Comparison of training performance of our deblurring network with and without local-residual connections.

S3. Quantitative Evaluation on static scenes

In Table 1, we had evaluated our method on blurred images of dynamic scenes from [23], and presented comparisons with 5 state-of-the-art deep learning based methods [33,7,23,18,34]. The main application of our work is efficient extraction of motion and sharp content from such general dynamic scenes. Due to the complexity of the blur present in such images, conventional image formation model based deblurring approaches struggle to perform well. Hence, our comparisons included only 2 conventional methods ([39,42] were selected as representative traditional methods for non-uniform deblurring, with publicly available implementations).

In Table S2, we provide quantitative evaluation on the static-scene blurred dataset of [19] along with comparisons with conventional approaches of [3,5,17,25] (whose de-

Table S1. Architecture details of our proposed Deblurring Module (DM). The symbol $(+k)$ denotes the width increment on the densely connected path.

stage	output	Layer Details
	$\frac{H}{2} \times \frac{W}{2}$	Space To Depth Transformation Factor of 2
Conv1	$\frac{H}{2} \times \frac{W}{2}$	$7 \times 7, 12, 64$, stride 1
RDB1	$\frac{H}{2} \times \frac{W}{2}$	$\begin{bmatrix} 3\times3,64, GR \\ 3\times3,64+GR, GR \\ 3\times3,64+2*GR, GR \\ 3\times3,64+3*GR, GR \\ 3\times3,64+4*GR, 64 \end{bmatrix} \times 3$
Conv2	$\frac{H}{4} \times \frac{W}{4}$	$3 \times 3, 64, 96$, stride 2
RDB2	$\frac{H}{4} \times \frac{W}{4}$	$\begin{bmatrix} 3\times3,96, GR \\ 3\times3,96+GR, GR \\ 3\times3,96+2*GR, GR \\ 3\times3,96+3*GR, GR \\ 3\times3,96+4*GR, 96 \end{bmatrix} \times 3$
Conv3	$\frac{H}{8} \times \frac{W}{8}$	$3 \times 3, 96, 128$, stride 2
RDB3	$\frac{H}{8} \times \frac{W}{8}$	$\begin{bmatrix} 3\times3,128, GR \\ 3\times3,128+GR, GR \\ 3\times3,128+2*GR, GR \\ 3\times3,128+3*GR, GR \\ 3\times3,128+4*GR, 128 \end{bmatrix} \times 3$
Bottleneck1	$\frac{H}{8} \times \frac{W}{8}$	$\begin{bmatrix} 1\times1,128, 128*2 \\ 3\times3,128*2, 128 \end{bmatrix}$
Deconv1	$\frac{H}{4} \times \frac{W}{4}$	$3 \times 3, 128, 96$, stride 2
Bottleneck2	$\frac{H}{4} \times \frac{W}{4}$	$\begin{bmatrix} 1\times1,96, 96*2 \\ 3\times3,96*2, 96 \end{bmatrix}$
Deconv2	$\frac{H}{2} \times \frac{W}{2}$	$3 \times 3, 96, 64$, stride 2
Bottleneck3	$\frac{H}{2} \times \frac{W}{2}$	$\begin{bmatrix} 1\times1,64, 64*2 \\ 3\times3,64*2, 64 \end{bmatrix}$
Deconv3	H × W	$3 \times 3, 64, 32$, stride 2
Projection Conv1	$\frac{H}{4} \times \frac{W}{4}$	$3 \times 3, 96, 96$, stride 1
Projection Conv2	$\frac{H}{2} \times \frac{W}{2}$	$3 \times 3, 64, 64$, stride 1
Conv4	H × W	$3 \times 3, 35, 16$, stride 1
Conv5	H × W	$3 \times 3, 16, 3$, stride 1
# params (for GR=32)		$\mathbf{4.46} \times 10^6$

blurred outputs are publicly provided by [19]) and the recent method of [36]. The set consists of 100 non-uniformly blurred images of scenes containing people, faces, text, saturated regions, natural and man-made structures. Quantitative comparisons (computed based on best alignment) reveal the parity/superiority of our method on this deblurring dataset as well, since our approach is more successful in capturing spatially varying nature of blur and delivers artifact-free results. An additional highlight of our method is that it is quite fast and does not involve any parameter tuning during test phase.

Table S2. Quantitative comparisons on Lai's Dataset [19].

Method	[5]	[3]	[17]	[25]	[39]	[42]	[36]	Ours
PSNR	16.71	17.98	17.90	18.47	18.41	18.43	18.94	**18.97**
SSIM	0.675	0.733	0.738	0.759	0.719	0.750	0.768	**0.773**

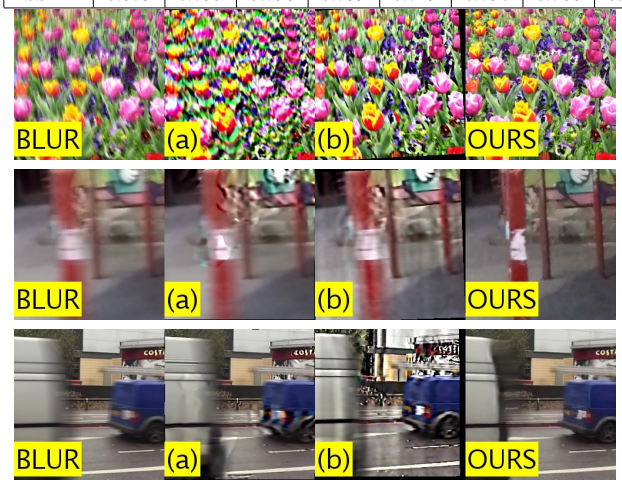


Figure S7. Comparison of video extraction results of our method with [7] and [36].

S4. Comparisons with Existing Motion Extraction Approaches

We also compare our video estimation approach with existing methods of [7] and [36], that estimate motion from a blurred image as an intermediate step for motion deblurring. Although these methods are proposed purely for the purpose of deblurring, we utilize their deblurred image and motion trajectory to construct a video from a single blurred image. Fig.S7 shows the comparison of generated videos of the three approaches for 3 scenarios: a fronto-parallel scene (1^{st} row), 3D scene (2^{nd} row) under pure camera motion, and a dynamic scene (3^{rd} row).

The authors of [7] proposed to deblur dynamic scenes by estimating a single motion flow-map from a given blurred image. To obtain a trajectory, we interpolated along the predicted flow map to get 9 motion flows which are then applied on their deblurred image to get a video. It can be observed that results of [7] (in Fig.S7(a)) suffer from severe inconsistencies in pixel motion for scenes containing moderate blur since [7] only encodes short range motion. In contrast, our method's results are more accurate and realistic.

The method of [36] estimates a motion density function (MDF) which determines the relation between a blurred image and the corresponding sharp image. As mentioned in

the main paper, the poses in MDF do not have a time-stamp associated with them. It is non-trivial to arrive at a temporal ordering of these poses. Importantly, unlike our model, the MDF by design addresses only camera motion and cannot handle independent object motion or 3D scenes. Nevertheless, we compare our video extraction results with the best possible videos which can be constructed with their outputs.[1] To extract a motion trajectory from MDF, we fit a 4^{th} order polynomial through MDF poses and sampled 9 points along the trajectory. These 9 camera poses are used to warp their deblurred image to obtain 9 frames. As shown in Fig.S7(b)), while the MDF result appears plausible for the constrained case of fronto-parallel static scene, it fares poorly for general scenes where the generated videos contain the same motion for the entire scene (as expected), which is quite inconsistent with the blur in the input image. In contrast, our results are more realistic and faithful to the blurred image.

S5. Additional Qualitative Results for Video Extraction

Results on GoPro dataset [23]: In Figs. S8-S10 we provide additional results of video extraction on test images constructed using videos from the test set of GoPro dataset [23]. These results demonstrate our network's ability to handle 3D scenes with dynamic object motion and camera motion. It can be observed our results closely mimic the ground-truth videos, while the results of [13] suffer from artifacts even when the corresponding regions contain only a moderate amount of blur. Such differences become more pronounced on images affected with large blur (blurred images created by averaging more than 7 frames), as shown on two examples in Fig. S11.

Results on Blur Detection dataset [32]: In Figs. S12-S16, we provide additional results of our 7 frame model on images from the dataset of [32] which contains a wide variety of real blurred images.

The comparisons demonstrate that in many cases, the results of [13] suffer from local motion and color distortions (Fig. S12), failure in detecting motion (Fig. S13) as well as distortions and inconsistencies due to deblurring (Figs. S14,S15). In cases containing mild blur (Fig. S16), their results are comparable to ours. Note that by having a single recurrent network to generate the video, our network can be directly trained to extract even higher number of frames (> 9) without any design change or additional parameters. In contrast, [13] requires training of an additional network for each new pair of frames.

[1]We used the implementation provided by the authors of [36] upon request.

S6. Additional Qualitative Results for Single Image Deblurring

In Figs. S17-S24, we provide visual comparisons of our results for single image deblurring with existing deblurring approaches on diverse scenes from the GoPro dataset [23]. In comparison to the results of existing methods, the texture details in our results are much closer to the ground-truth sharp frame.

S7. Discussion

We addressed an interesting problem in that blurred images have generally been considered as a nuisance and the usual practice is just to deblur the image. In contrast, our work reveals the richness of information embedded within a blurred image that can be ably derived to convey how the camera and objects in a scene move. Our complementary networks extract pixel-level motion which can potentially be utilized to gain insights into the nature of incidental ego-motion, distinguish between static and dynamic contents in a scene, reveal sub-pixel motion, and yield plausible temporal ordering.

The prime reason behind the effectiveness of our framework is decomposition of the task into multiple sub-problems. Specifically, modeling multi-frame estimation task as recurrent motion prediction extracts scene dynamics while preserving scene appearance. Our convolutional recurrent design for motion prediction greatly improved training efficiency of RVE-RVD (took only 6 hours / 5×10^4 iterations to converge). Our choice of video reconstruction as a proxy task (less ambiguous and simpler) improved our network's convergence.

Our approach could potentially be utilized in several other vision tasks. The motion flows estimated by BIE-RVD can be utilized for blur-based dynamic region segmentation, our unsupervised motion prediction framework (BIE-RVD) can be extended to perform depth estimation for static scenes where the spatially varying blur is linked to the depth of the scene, and the motion encoding extracted by our BIE can be used for action recognition from a single blurred image.

Figure S8. Comparisons of our video extraction results with [13] on motion blurred images created from the test videos of [23]. In the top-down order, we show the blurred input, result of [13], our result and the ground-truth video.

Figure S9. Comparisons of our video extraction results with [13] on motion blurred images created from the test videos of [23]. In the top-down order, we show the blurred input, result of [13], our result and the ground-truth video.

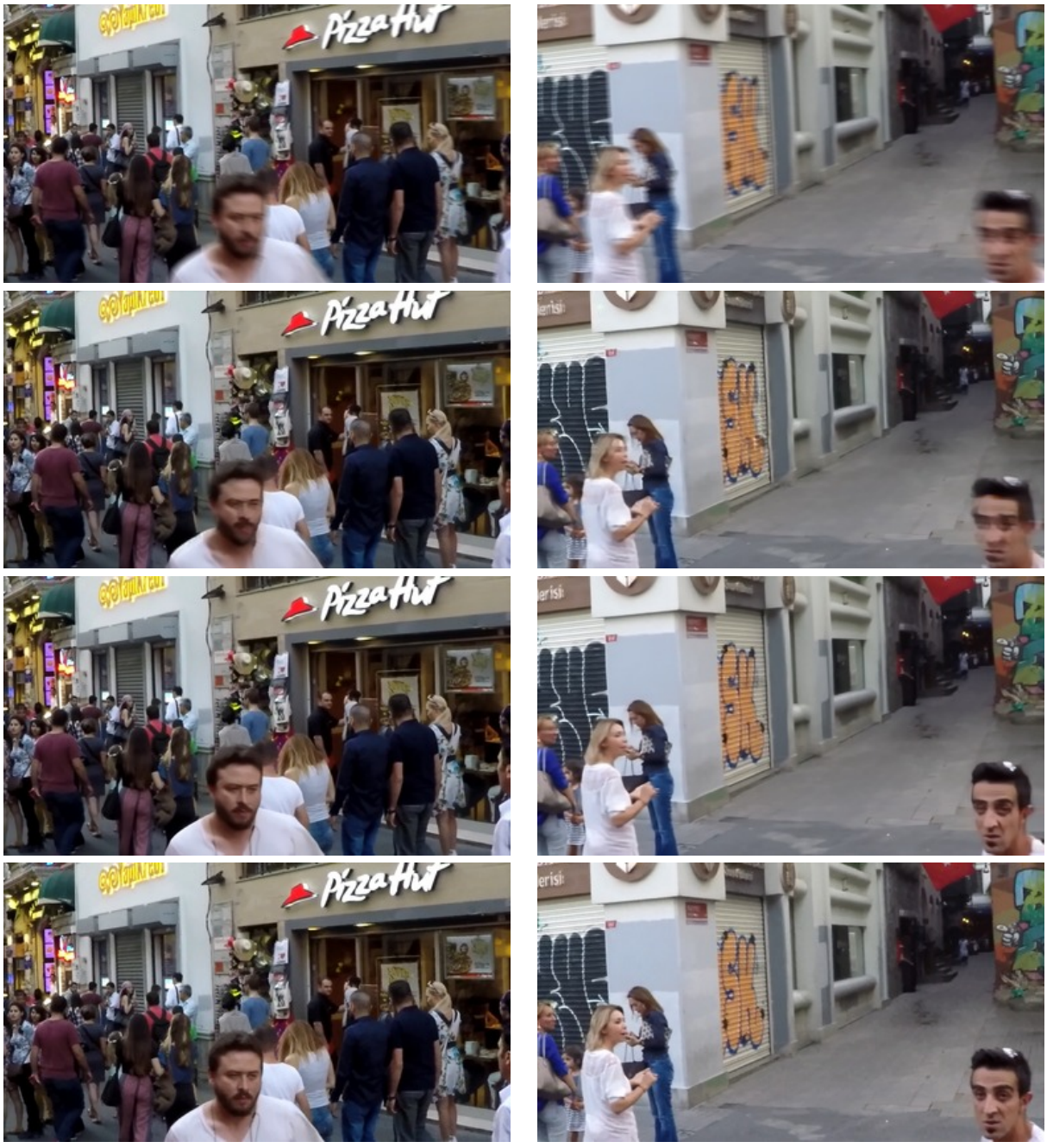


Figure S10. Comparisons of our video extraction results with [13] on motion blurred images created from the test videos of [23]. In the top-down order, we show the blurred input, result of [13], our result and the ground-truth video.

Figure S11. Comparisons of our video extraction results with [13] on motion blurred images from the test videos of [23]. In the top-down order, we show the blurred input, result of [13] and our result.

Figure S12. Video extraction results on dataset of [32]. The left column contains input motion blurred images while the generated videos using [13] and our method are shown in the middle and the right column, respectively. These results show the local motion and color distortions in videos generated by [13]. Results using [13] have distortions on the train door in Example 1, road in Example 2, basketball hoop in Example 3 and road in Example 4.

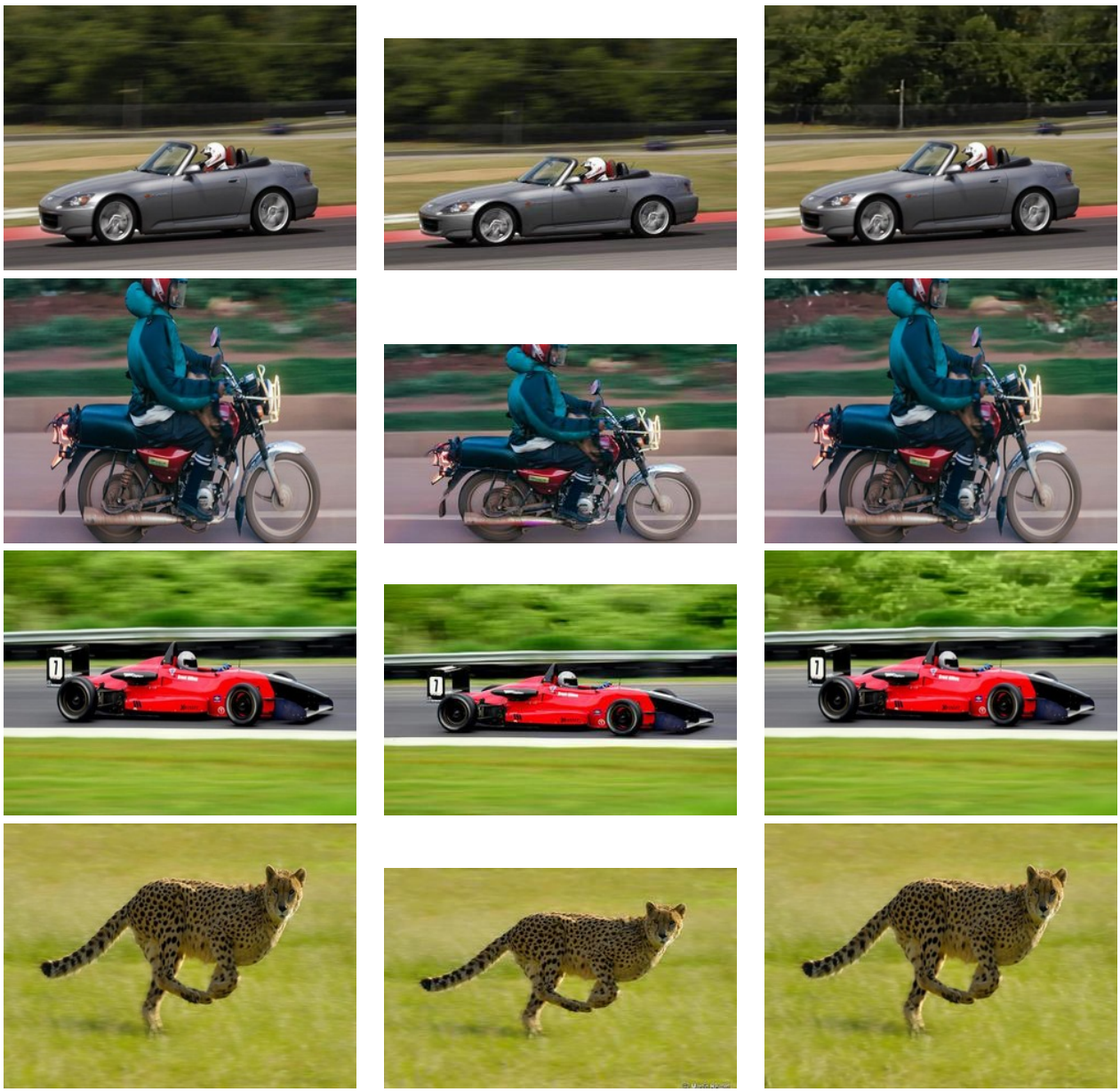


Figure S13. Additional results on the dataset of [32]. These results show some cases where method [13] fails in detecting motion.

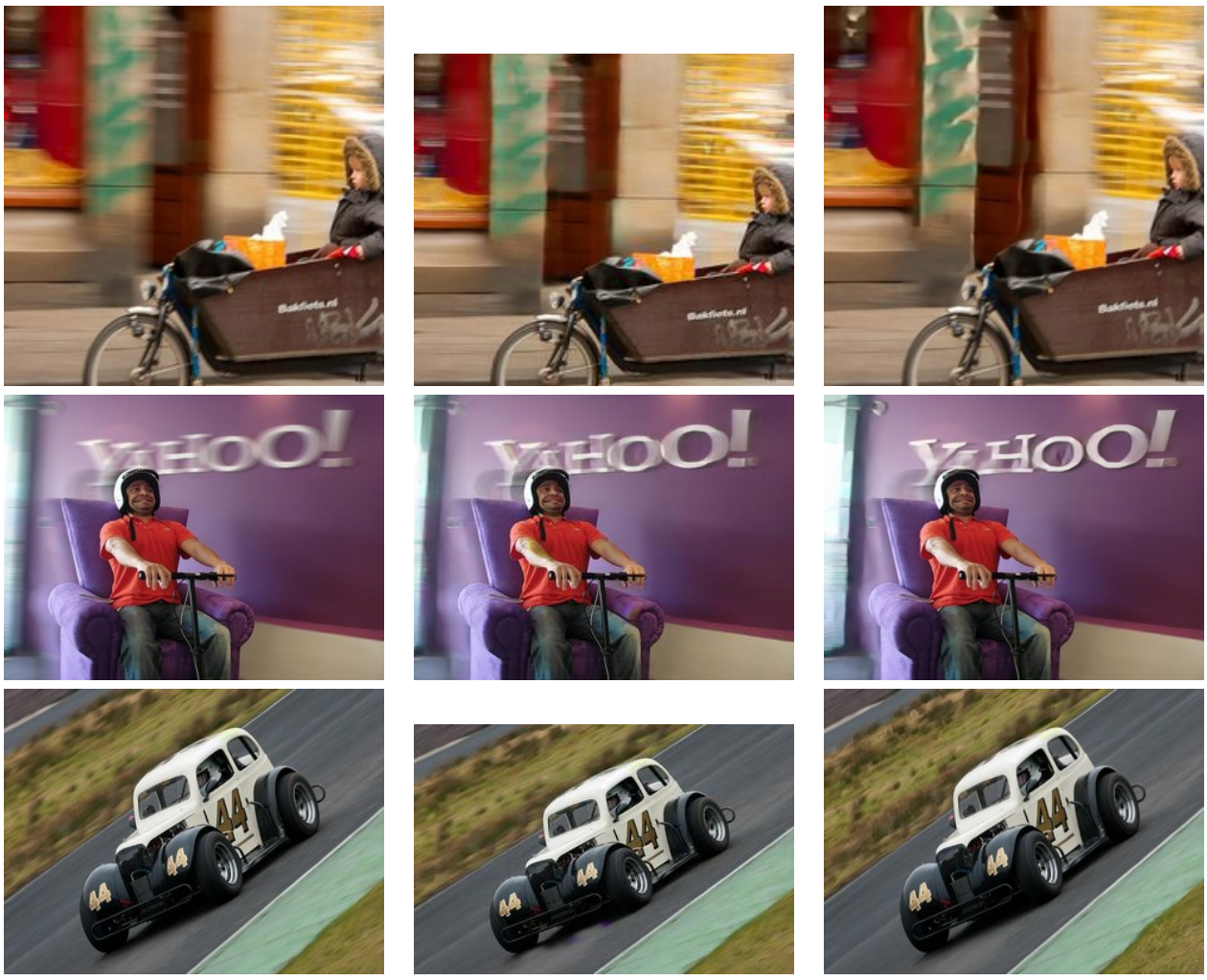


Figure S14. Additional results on the dataset of [32]. Results using [13] have distortions near green paint in Example 1, arm-rest in Example 2 and road in Example 3

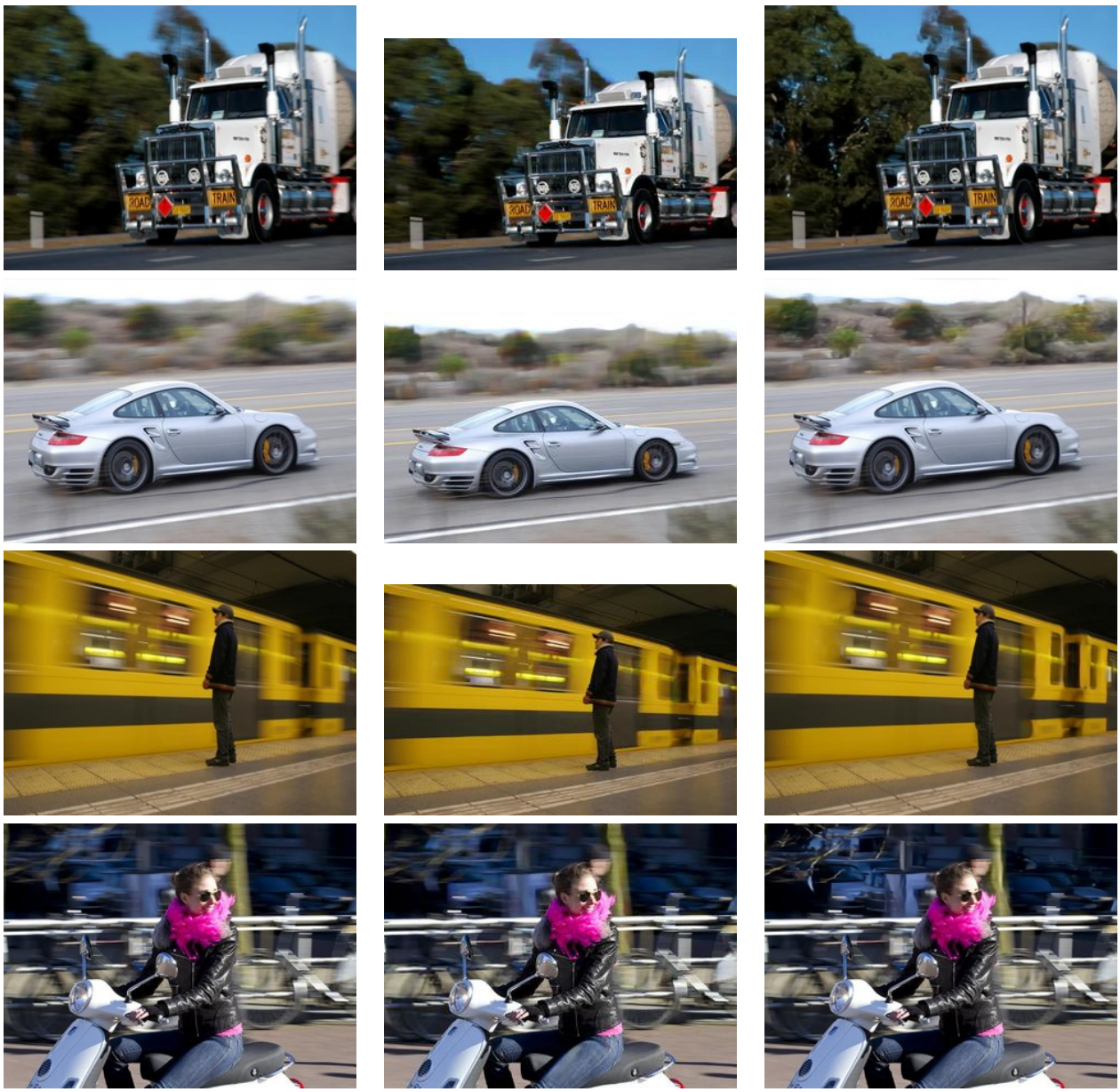


Figure S15. Additional results on the dataset of [32]. These results show distortions and inconsistencies in videos generated by [13].

Figure S16. Additional results on the dataset of [32]. These are results on images having mild blur and results of [13] are comparable to ours.

Figure S17. Visual comparison for deblurring on images from GoPro test-set. The figure shows the full sized images along with zoomed-in patches corresponding to the Blurred image, results of [39], [23], [18], [34], Ground-truth and Our Result.

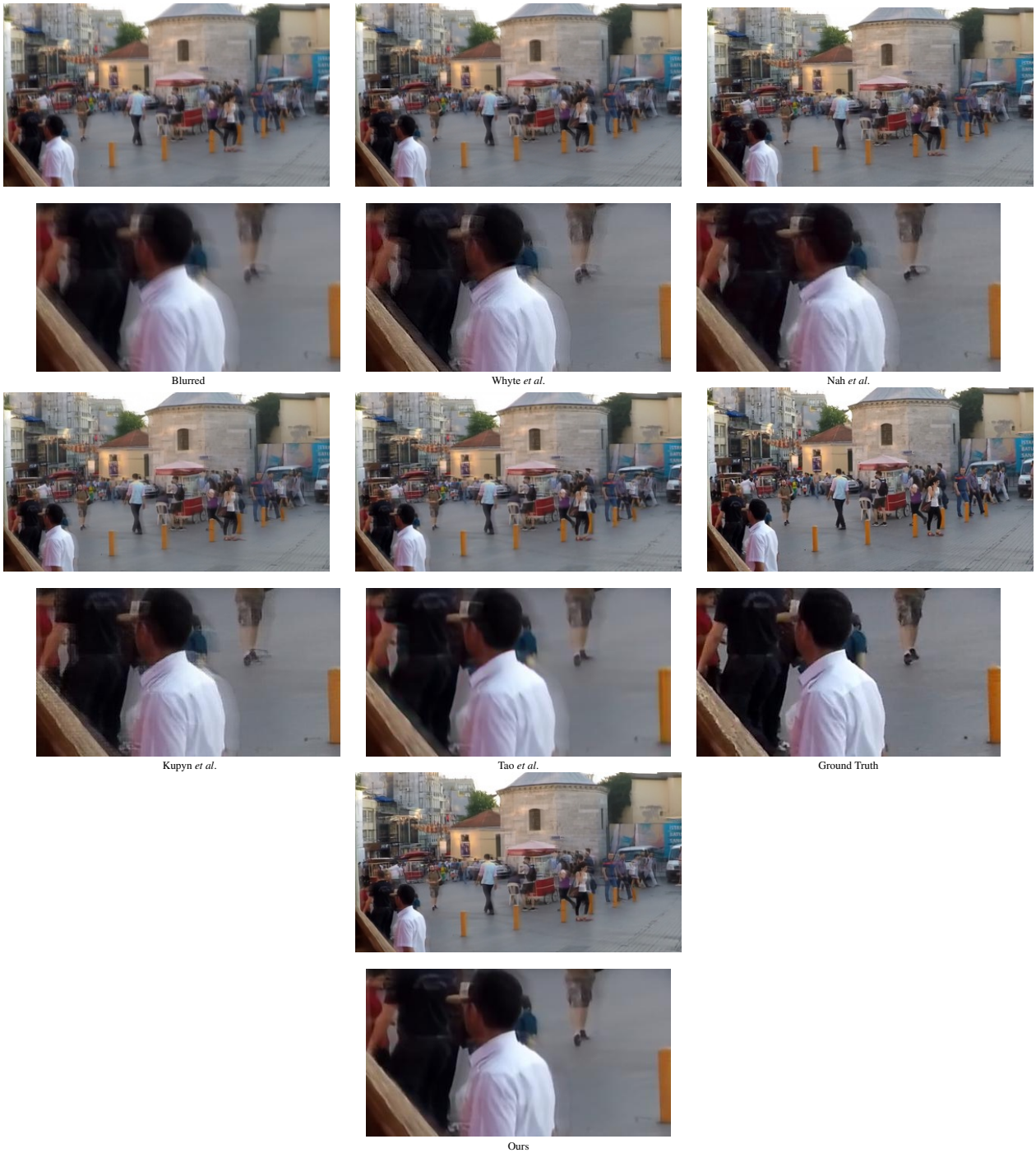


Figure S18. Visual comparison for deblurring on images from GoPro test-set. The figure shows the full sized images along with zoomed-in patches corresponding to the Blurred image, results of [39], [23], [18], [34], Ground-truth and Our Result.

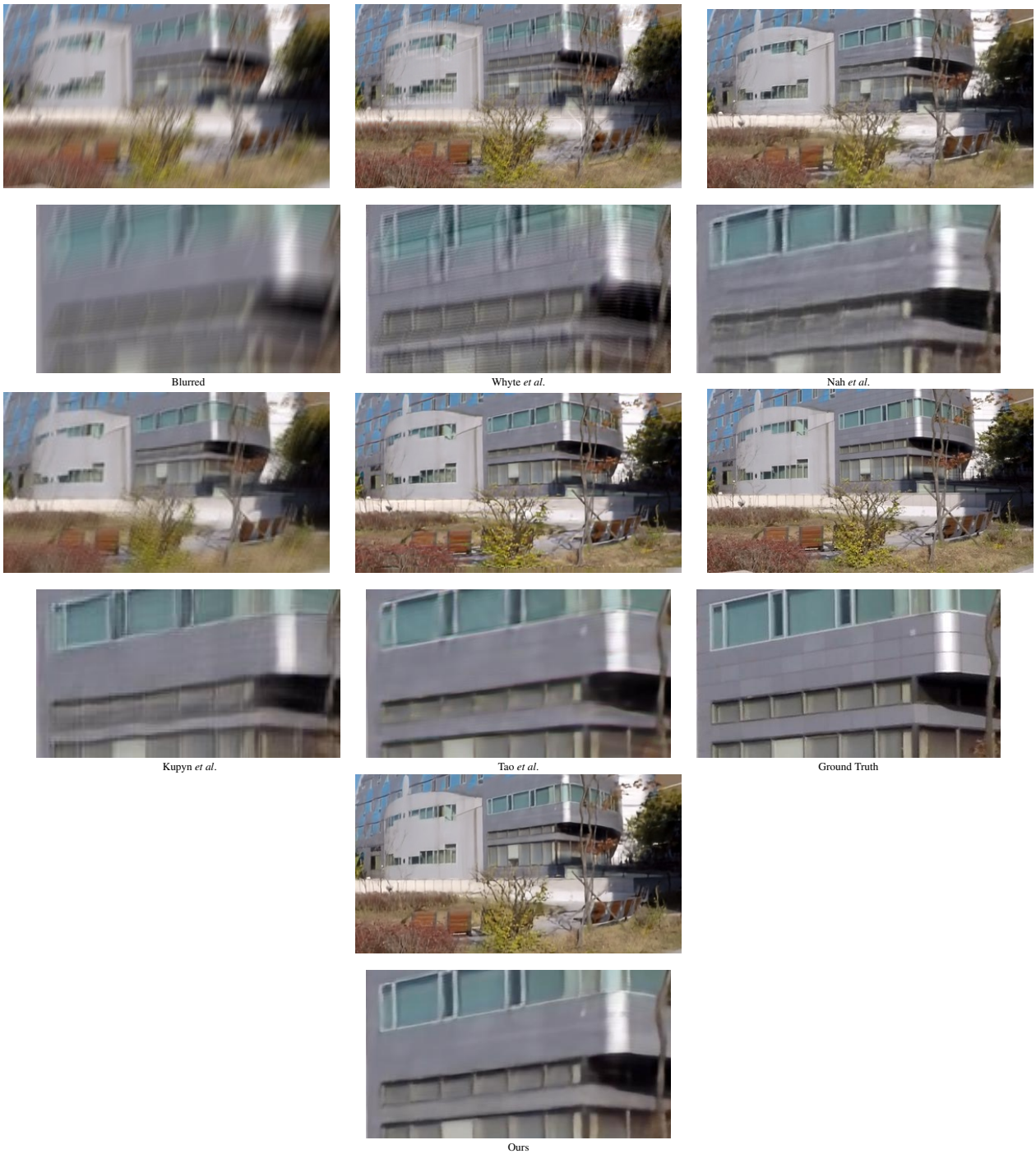


Figure S19. Visual comparison for deblurring on images from GoPro test-set. The figure shows the full sized images along with zoomed-in patches corresponding to the Blurred image, results of [39], [23], [18], [34], Ground-truth and Our Result.

Figure S20. Visual comparison for deblurring on images from GoPro test-set. The figure shows the full sized images along with zoomed-in patches corresponding to the Blurred image, results of [39], [23], [18], [34], Ground-truth and Our Result.

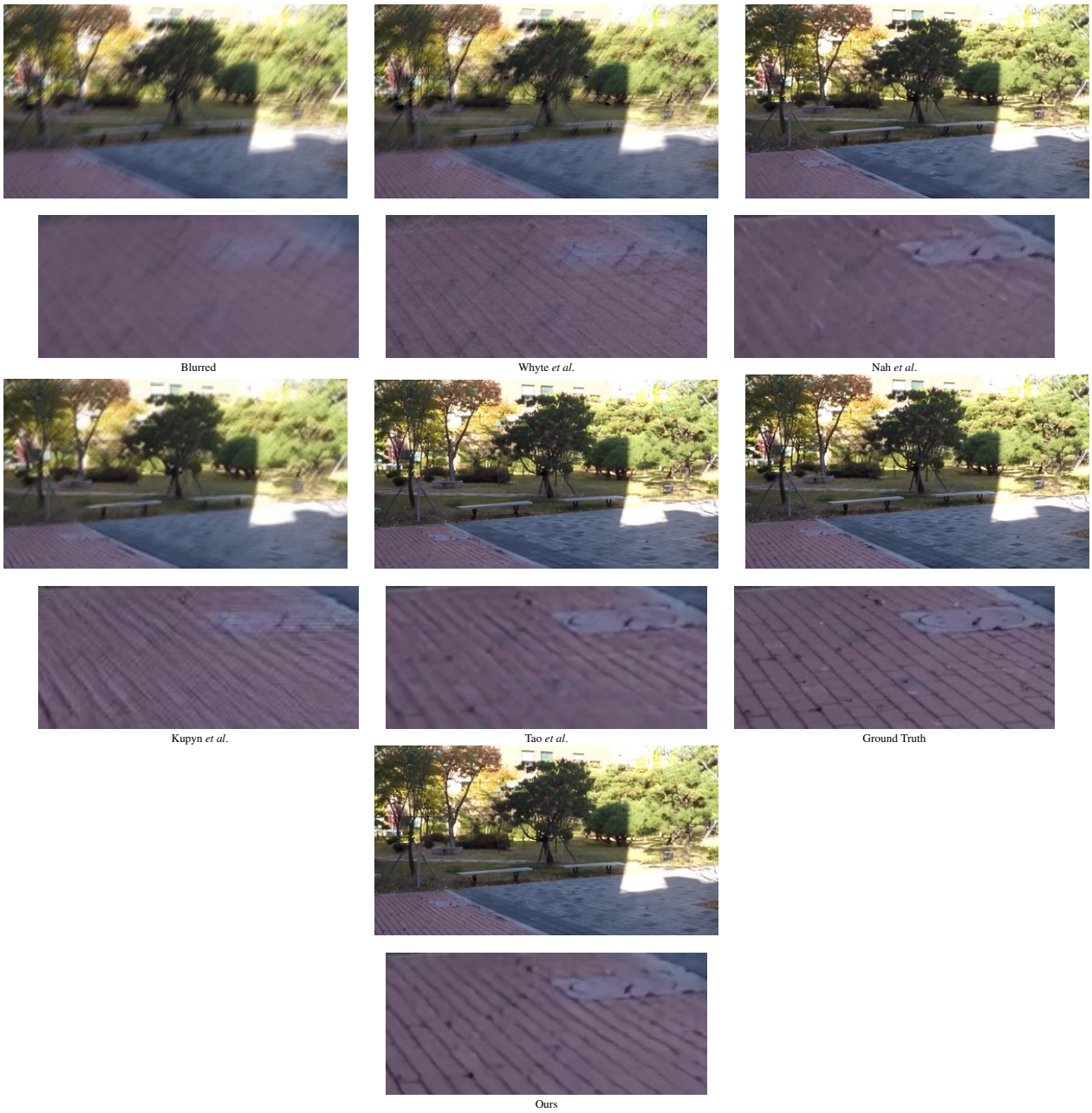


Figure S21. Visual comparison for deblurring on images from GoPro test-set. The figure shows the full sized images along with zoomed-in patches corresponding to the Blurred image, results of [39], [23], [18], [34], Ground-truth and Our Result.

Figure S22. Visual comparison for deblurring on images from GoPro test-set. The figure shows the full sized images along with zoomed-in patches corresponding to the Blurred image, results of [39], [23], [18], [34], Ground-truth and Our Result.

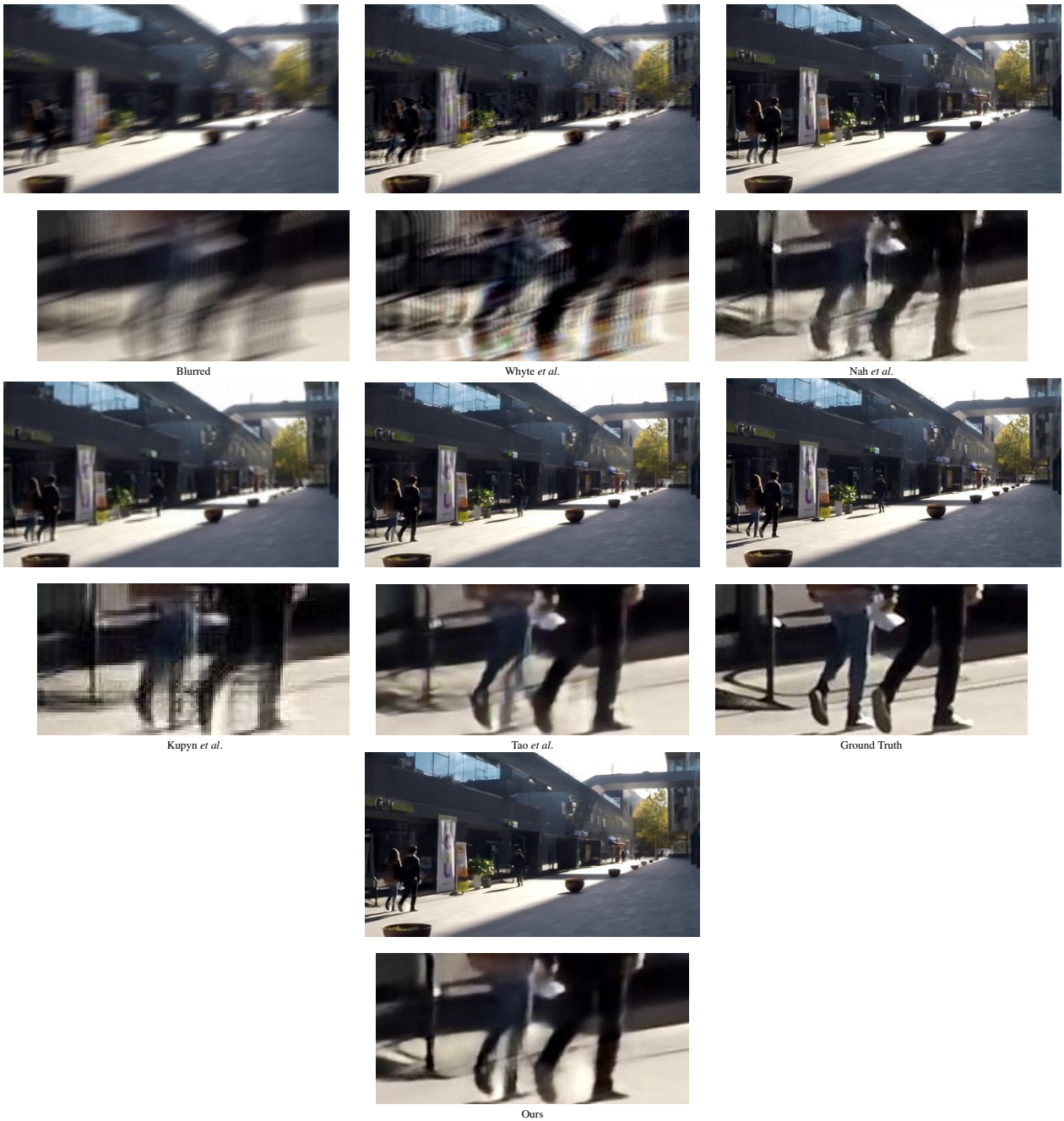


Figure S23. Visual comparison for deblurring on images from GoPro test-set. The figure shows the full sized images along with zoomed-in patches corresponding to the Blurred image, results of [39], [23], [18], [34], Ground-truth and Our Result.

Figure S24. Visual comparison for deblurring on images from GoPro test-set. The figure shows the full sized images along with zoomed-in patches corresponding to the Blurred image, results of [39], [23], [18], [34], Ground-truth and Our Result.