Improved Handling of Motion Blur in Online Object Detection

Mohamed Sayed   Gabriel Brostow
University College London
visual.cs.ucl.ac.uk/pubs/handlingMotionBlur/

Abstract

We wish to detect specific categories of objects, for on-line vision systems that will run in the real world. Object detection is already very challenging. It is even harder when the images are blurred, from the camera being in a car or a hand-held phone. Most existing efforts either focused on sharp images, with easy to label ground truth, or they have treated motion blur as one of many generic corruptions.

Instead, we focus especially on the details of egomotion induced blur. We explore five classes of remedies, where each targets different potential causes for the performance gap between sharp and blurred images. For example, first deblurring an image changes its human interpretability, but at present, only partly improves object detection. The other four classes of remedies address multi-scale texture, out-of-distribution testing, label generation, and conditioning by blur-type. Surprisingly, we discover that custom label generation aimed at resolving spatial ambiguity, ahead of all others, markedly improves object detection. Also, in contrast to findings from classification, we see a noteworthy boost by conditioning our model on bespoke categories of motion blur.

We validate and cross-breed the different remedies experimentally on blurred COCO images and real-world blur datasets, producing an easy and practical favorite model with superior detection rates.

1. Introduction

A little motion blur is present in most hand-held photography. Blur is ever harder to ignore because images are increasingly captured on the move, *e.g.* by a gimbaled robot or from an autonomous vehicle. Precisely these on-the-go situations prompt us to explore: how much does motion blur severity impact object detection? What can be done about it? Detection is important because it underpins many other tasks, such as tracking and re-identification, and our initial scope is further narrowed to egomotion induced blur.

Unsurprisingly, the severity of the blur correlates with detection failure [2]. Fig. 1 shows an example. An ideal algorithm will make that degradation more gradual, and could

Figure 1. a) Original sharp MS COCO [26] image with object detections. b) Same image with significant linear motion-blur, with COCO ground-truth. c) Failed predictions from original Faster-RCNN. d) Predictions from network with our proposed model.

someday enable a model that surpasses even a human's ability to see through blur. Instead of a single breakthrough, it is more likely that a combination of approaches is needed. Much like the "devil in the details" papers [6, 7], the task specifics and pipeline likely make a difference.

Our main contribution is an empirical exploration of five classes of remedies. These remedies are selected to cope with five proposed causes for reduced detection accuracy. The five cause/remedy pairs explored here are: **1)** *Is the entire image too blurry to be useful?* Deblur test image first. **2)** *Is texture mismatch along blur axes confusing the model?* Spatially transform image to compensate. **3)** *Does test-time blur differ from training data?* Train model for out-of-distribution robustness, and/or perform test-time tuning of network. **4)** *Are the training labels incorrect?* Customize labels to match detection-in-blur task and reconsider labels used for testing. **5)** *Are egomotion blur types too diverse?* Treat detection in blur as a multi-task problem.

Overall, we propose a new model that focuses on the remedies from (4) and (5), and set a new standard for online object detection in egomotion-induced blur.

2. Related Work

Deblurring: A close topic with valuable data and potential insights is image deblurring. The first canonical method
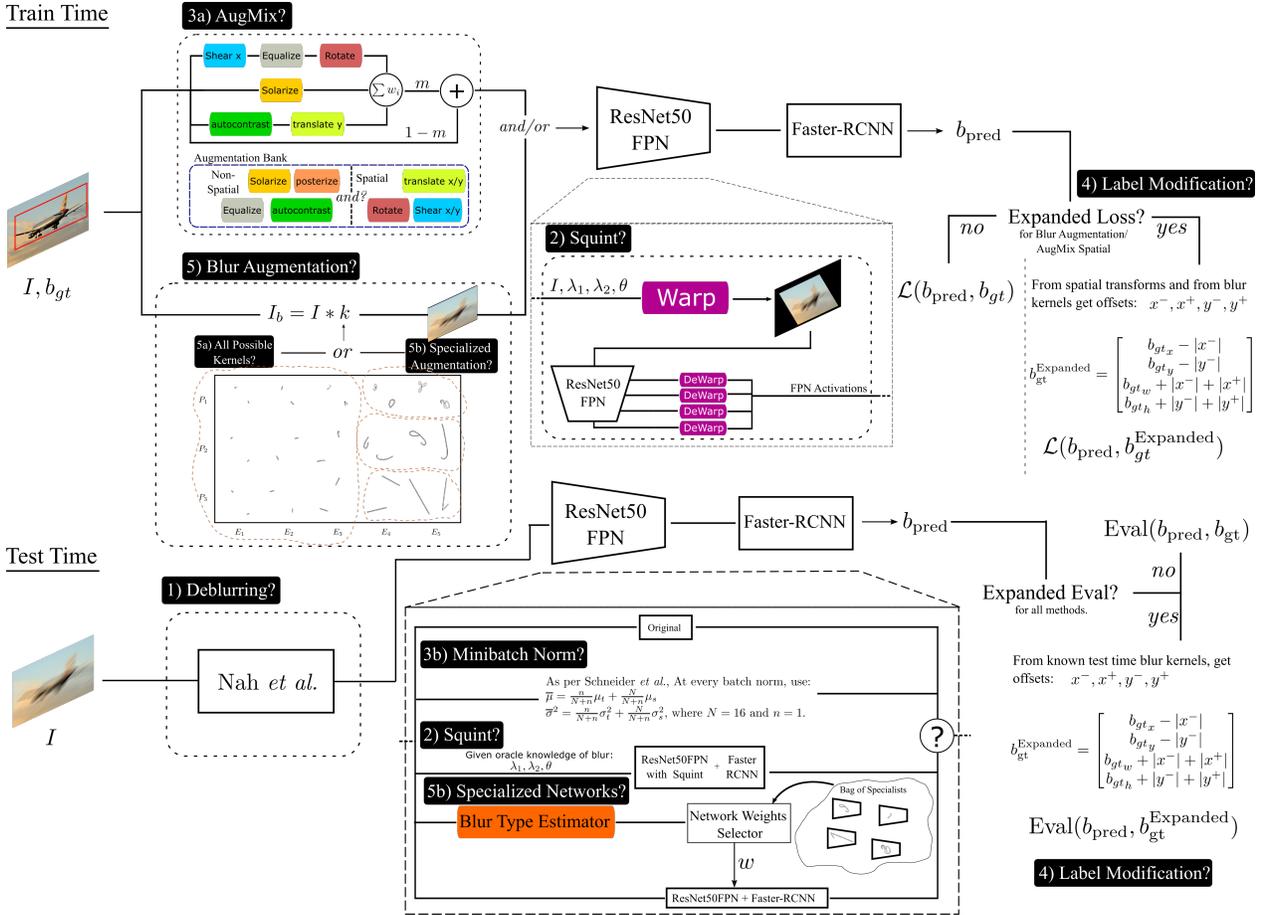
Figure 2. We explore the baseline with five categories of remedies across both train and test time. **1) Deblurring:** Deblur the input using Nah *et al.* [31] before passing it to the detector at test time. **2) Squint:** Influenced by [19, 37], we undersample the incoming image on each axis, according to known blur kernel shape during training and testing (assuming oracle knows test kernel). We forward pass through the backbone and carry out the reverse sampling operation on outgoing activations, before passing them to the detection head. **3) Out-of-distribution:** Treat motion blur as an out-of-distribution robustness problem and use two leading methods as remedies: **3a) AugMix:** Augment training images as in AugMix [16] with two main flavors: all augmentations from the AugMix paper, or only non-spatial augmentations. We then use the model normally at test time. **3b) Using minibatch statistics:** As in [43], modify the batch normalization statistics at every layer in the network to be a weighted average of the incoming minibatch (size one) statistics and the original training-averaged batch normalization statistics. **4) Label Modification** Under blur, the spatial extent and center of the bounding box are ambiguous. We experiment with training with labels expanded to include the superset of object locations given by the extent of the blur kernel's reach, under blur augmentation and when using spatial augmentation in AugMix. We also report results for both expanded COCO minival labels and without. **5) Blur Augmentation** Augment COCO images using either **5a)** a random selection of kernels across all possible motion-blurs we can generate, or **5b)** training networks specialized on specific blur-kernel varieties. At test time, we use either a network trained under general blur augmentation, or a system that incorporates a bag of models and a blur estimation module that selects the appropriate specialized network for the task.

for image deconvolution comes from Richardson [39] and Lucy [28] where a known point spread function (PSF) - the blur kernel - is used to iteratively minimize an energy function to find a maximum likelihood estimate of the original image. Deblurring can be non-blind where the blur kernel is known [42], or it can be blind, where the kernel is either first estimated [11, 44, 23] - usually optimized with the final result [8] - or the entire deblurring method is non-interpretable

and runs end to end [31, 49]. Deblurring can also assume either a uniform blur kernel throughout an image [11, 44] or variable nonuniform blur either due to camera egomotion (rotation, zoom) [52], depth-of-field effects [48], or dynamic object motion blur [18, 31, 23].

Previous work has made use of an L0 sparse representation [53], dark image regions [32], and multiple frames in a video [45]. More invasive methods exploit hard-

ware, including using a coded shutter [36], inertial measurements [22], flash frame information integration [57], bursts of blurry images [1], high and low frame rate cameras [47], or an event driven camera tied to an RGB sensor [33].

Some deep learning deblurring methods are interpretable [44, 5, 54, 46], but most are end-to-end [49, 24, 55, 31] with the recent state-of-the-art by Nah *et al.* trained on the high frame rate GOPRO dataset [31]. We explore using a state-of-the-art deblurring method as a preprocessing step, and measure the overall effectiveness of such a baseline.

Boracchi *et al.* [3] generate a statistical model for motion blur kernel generation for benchmarking image restoration model performance. The blur kernels they generate are parameterized to simulate camera shake and exposure, making their kernel generation method a good candidate for synthesized blur augmented training.

Although deblurring's aesthetics driven approach means that there are competitive methods for extracting high frequency information in blur, use in an online vision application might be impractical, especially given that networks are very sensitive to changes in training distribution.

Blur and Scene Understanding Tasks: Directly related to this work, Vasiljevic *et al.* [51] explore the effect of blur on ImageNet [10] classification performance and blur augmentation strategies using a set of synthetically generated blur kernels; however they use a limited set of 100 17×17 pre-generated fixed length motion blur kernels and a restrictive image resolution of 384×384 during training and evaluation. They experiment with different blur types and fine grained blur augmentation for classification, but only a segmentation dividing blur types - and not across blur exposure with different kernel types - is considered; only defocus blur is explored for image segmentation. For image segmentation, they evaluate their networks using a soft boundary for accuracy, but do not explore the effect of spatial ambiguity on fine-tuning networks for blur during training, especially since only defocus blur (no shift in barycenter naturally) is used for fine-tuning the segmentation task. Vasiljevic *et al.* note that knowing blur information apriori could be helpful, but don't explore such a blur estimator.

Overall, we find that building explicit robustness into vision models for dealing with realistic camera motion blur needs more exploration, especially for spatial tasks.

Out-of-Distribution Robustness: Recent work [15, 14] treats image corruptions (brightness, contrast, snow, noise, blur) as out-of-distribution samples compared to the in-distribution clean images a network was trained on. ImageNet-C [15] is a variant of the ImageNet classification dataset that contains images corrupted by 15 different types of canonical image corruptions, and is used as a benchmark for out-of-distribution model performance. Crucially, ImageNet-C - and others such as ImageNet-R [14] and ImageNet-A [17] - are not meant to be trained on. In-

stead, the argument is that a model's ability to generalize to images outside of the training set's distribution can be measured by evaluating its performance on these datasets. Although ImageNet-C contains motion blur corruptions, the method only considers straight line motion blur kernels. Michaelis *et al.* [29] use the same corruptions from ImageNet-C to produce a robustness benchmark for detection, by augmenting MS COCO [26]. COCO-C also includes straight line blurred images, but changes in labels under the spatial ambiguity brought upon by blur are not addressed. We call this type of naive blur 'Non-Centered,' and we show why it's important for spatial reasoning.

In the data augmentation space, AutoAugment [9] finds an optimal augmentation policy for a model and dataset pair achieving state-of-the-art accuracy for classification datasets, but requires 15,000 compute hours on an NVIDIA Tesla P100 for training ImageNet. Rusak *et al.* [41] propose an adverserial noise training scheme for increasing classification model accuracy and robustness on ImageNet-C, mainly combating pixel noise and not blur. AugMix [16] is an augmentation strategy for improving classification model robustness to out-of-distribution images. It involves alpha blending copies of a training image that have been corrupted by a random chain of image augmentations. They use the same corruptions in [9], including both pixel level value changes and spatial augmentations. Although the AugMix paper also doesn't discuss how spatial augmentation should affect spatial labels, we explore the effectiveness of AugMix for blur robustness after making decisions on how spatial labels should be changed.

Schneider *et al.* [43] analyze the effect of normalizing activations in batch normalization layers using a weighted average of the statistics of both the source training set and the minibatch. Their method achieves state-of-the-art on ImageNet-C and improves ImageNet-C robustness on vanilla Resnet-50 classification models, even with a minibatch of size one.

While these methods are a promising way of increasing model robustness to unseen corruptions, the aim of the proposed work is to explore the specific impact of motion blur on detection, and so we focus our effort on manufacturing the most realistic blur kernels available in the literature.

3. Designing Detection Models for Motion Blur

To improve online object detection, we propose a unified framework that allows us to measure the impact of different remedies and their combinations. The framework is based on a state-of-the-art object detector, Faster-RCNN [38], with training and testing on data derived from the MS COCO [26] detection dataset. The baseline and data are explained in this section. The proposed remedies are explained in detail in Sec. 4, and are evaluated in Sec. 5. Figure 2 illustrates both the baseline model, and the different

enhanced alternatives.

3.1. Detection Baseline

For reproducibility, we use the pretrained Faster R-CNN variant trained on COCO, available through Pytorch's torchvision [34] library, as a baseline for all our experiments. We use a ResNet-50 [13] backbone with a Feature Pyramid Network (FPN) [25]. This baseline achieves 58.5 mAP@0.5 and 37.0 mAP@0.5:0.95 on the COCO test set. While other models achieve better accuracy on the COCO minival set, we choose this framework for its accessibility and as a good baseline representation of a canonical detection framework with top-10 performance for the backbone's size [21, 27].

3.2. Selecting Data for Training and Testing

Ideally, we'd select data with detection labels for images exhibiting motion blur. Due to the way MS COCO is gathered [26], there are very few blurry images in the dataset. This leaves us with the task of generating synthetically blurred COCO images for both training and evaluation. Related but not directly applicable here, there are multiple real-world image datasets for deblurring. These were generated using high frame rate video [31] or shutter tied cameras [40]. They either don't contain enough images for training and evaluating detection models ([40] only contains 5500 images) and/or lack object annotations. Zhang *et al.* [56] generate blurry images as part of a GAN architecture for deblurring. Although they train the blur generation module using a discriminator trained on real world blurry images, it is not trivial to modify labels, given spatial ambiguity, since camera motion is not made explicit. Brooks & Barron [4] use multiple adjacent images (as few as two) to generate realistic motion blur. But to use that would require a video or stereo dataset with ground truth labels for the detection task.

This leaves methods that synthesize blurry images via convolution with synthetic motion blur kernels [42, 3, 51, 29, 15]. ImageNet-C [15] and COCO-C [29] contain images blurred using straight line motion blur exclusively, with no control over simulated camera shake. Vasiljevic *et al.* [51] use a limited set of motion-blur kernels since they are constrained by a fixed length spline formation model. Boracchi & Foi [3] describe a method that allows control over different characteristics of a camera's trajectory through space, including the amount of shake and jerk with variable exposure.

3.3. Blur Generation and Space Discretization

We adapt the blur kernel generation method from Boracchi & Foi [3]. We fix their high level controllable parameter P to one of three values, P_{1-3}, representing three distinct
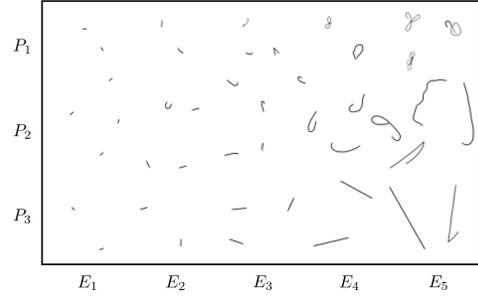

Figure 3. Example blur kernels based on Boracchi & Foi's [3] motion-blur model. All kernels occupy a space of 128×128. In our modification, used everywhere unless stated otherwise, kernels are centered by shifting the barycenter of all nonzero points to the origin of the filter. P_1 type kernels tend to be very erratic, while P_3 kernels contain mostly rectilinear trajectories representative of when camera ego motion is linear.

types of camera motion. We also modulate exposure via early camera trajectory clipping.

First, we generate a trajectory by finding a random path in 2D space. We assign an initial velocity vector $\mathbf{v_0}$, drawn at random from a unit circle, and a position in space $\mathbf{x_0}$ for the camera. At every step, the camera's velocity vector is updated by the acceleration vector,

$$\mathbf{\Delta v} = P(\mathbf{\Delta v_g} - I\mathbf{x_t}), \quad (1)$$

where $\mathbf{\Delta v_g}$ is random acceleration with elements drawn from $\mathcal{N}(0, \sigma^2)$. $I\mathbf{x_t}$ is an inertial tendency for the camera to stay where it is, and $P \in P_{1-3}$ is the high level anxiety parameter we fixed above. P_3 has the highest random velocity change on every step. Further, to model a camera jerk, with a randomly sampled indicator function, the acceleration update also includes a component equal to twice the current velocity vector in a random direction, so

$$\mathbf{\Delta v} = \mathbf{\Delta v} + 2P|\mathbf{v}|\mathbf{\Delta v_j}, \quad (2)$$

where $\mathbf{\Delta v_j}$ is sampled from the unit circle. Again, with a high P, there is a higher chance of a jerk happening and a shakier camera. When starting a trajectory, I, σ^2, and j are drawn once from uniform random distributions to increase variability under the same class of blur $P \in P_{1-3}$. Note that this leads to some overlap between kernels generated across different Ps.

In summary, the type of camera behavior falls into one of three classes: 1) P_1 simulates a very nervous camera, 2) P_2 for back and forth behavior, and 3) P_3 simulates mostly straight rectilinear motion-blur. To simulate exposure, we stop the motion path early using the exposure factor (trajectory length) E. We discretize exposure to one of 5 values, E_{1-5}. Examples of these kernels can be seen in Fig. 3. Subpixel interpolation produces kernels for convolving sharp images.

3.4. Implementation Details

Kernel Generation: To speed up training, a corpus of 12,000 blur kernels is generated for every pair of $\{P_{1-3}\} \times \{E_{1-5}\}$, for a total of 180,000 possible motion blur kernels. However, random kernels are generated on the fly during evaluation for each combination of blur type and exposure, with fixed seeds for reproducibility. Trajectory length is 96 and blur kernels fit in 128×128 filters.

Blurring: Unlike in [15, 51, 29], we don't resize images to a fixed size before blurring. Instead each image is convolved separately with reflection padding, to account for what would otherwise be real world data. We opt not to resize our blur kernels to match image size as a way of simulating changes in focal length. We implement sparse convolution on the GPU for applying blur kernels. As per Sec. 4.4, we make sure to center our motion-blur kernels by translating their barycenters to the center of the filters.

Training: All networks start from a base Resnet-50FPN pretrained on COCO. We use an FPN framework that outputs activations at four scales from the backbone. There was no apparent difference in blur augmented performance when training all five blocks *vs.* fixing the weights of the first two.

4. Proposed Remedies for Improved Detection

Suspecting specific underlying causes for the adverse effects of blur on detection, we now propose bespoke remedies. Where appropriate, some of the remedies are also crossbred, and experimental results appear in Sec. 5.

4.1. Deblurring as a Pre-process

Image deblurring is useful for aesthetic purposes, but could also aid other vision tasks. To test this remedy, we use the recent deblurring model from the GoPro dataset paper, [31], before passing the result to the detector. Deblurring is a slow process, by $12\times$ in this case, so heavy optimizations would be needed for an online robot.

4.2. Reconciling Texture Information With Scale

When motion-blur is biased to one major direction over another, it removes more high frequency information (and texture) in that direction. It is reasonable to expect a network is not natively designed for this imbalance. CNNs usually understand texture and shape information across multiple scales under the same aspect ratio, but we're also asking the network to deal with a texture imbalance along the blur kernel's major axis. Influenced by the work on Spatial Transformers [19] (which proved a slightly inferior baseline) and neural sampling layers [37], we instead undersample the incoming image along the principal components of the blur kernel. The reverse operation is carried out, using reciprocal scaling factors, on every activation output

from the backbone. This "Squint" process is done at both train and test time. For best-case testing, an oracle is assumed to know the blur kernel.

4.3. Training *vs.* Test Distribution

We consider treating complex motion-blur as an out-of-distribution corruption as in [15, 29, 16, 43], and use two promising methods from the OOD literature. We use Aug-Mix [16] as a training time remedy. We propose three flavors, the first is a purely pixel level version where we augment pixel intensities only. The second applies all spatial augmentations as well as suggested in [16], but does not reconcile the shifts in bounding box changes. The third is an "Expanded" version following Sec. 4.4, where we change COCO labels at train time to match the superset of where an object is shifted to across branches. AugMix roughly approximates blur when augmentations are selected that translate an image before concatenating with other branches.

Further, at test time, we use covariate shift adaptation from the upcoming [43]. The first step is to get a weighted average of the incoming activation statistics of the mini-batch ($n = 1$ for online inference) and the source statistics of the model where $N = 16$,

$$\overline{\mu} = \frac{n}{N+n}\mu_t + \frac{N}{N+n}\mu_s, \text{and} \tag{3}$$

$$\overline{\sigma}^2 = \frac{n}{N+n}\sigma_t^2 + \frac{N}{N+n}\sigma_s^2. \tag{4}$$

We then use these new normalization statistics for batch normalization in all network layers.

4.4. Customizing Labels

When an image is motion blurred, objects are no longer confined to the bounding boxes they had occupied in the sharp image. The objective may no longer be to estimate that original bounding box. See Fig. 4. We discuss two remedies for this problem, that apply when training under augmentation and for evaluation.

Kernel Centering The start point of a motion blur path corresponds to the exposure at $t = 0$. Any path that leads away from the center, as in Fig. 4(b), will offset the blurred version of the object in some direction, so the "ground truth" bounding box is no longer centered on the blurred object. This introduces a mismatch between the blurred input and its label. This mismatch is created in [15, 29], introducing label ambiguity and noise in training.

We center a kernel using a weighted average of the kernel's nonzero points. The aim is to have the detection framework learn to localize objects based on where they are, on average, during the exposure. This remedy is similar to how [40] aligned images from paired long/short exposure cameras to train for deblurring. In our case, the training loss is noisier when training on non-centered kernels, and the drop

a) original image   b) non centered blur kernel (naive)   c) centered blur kernel (standard)   d) bounding boxes expanded
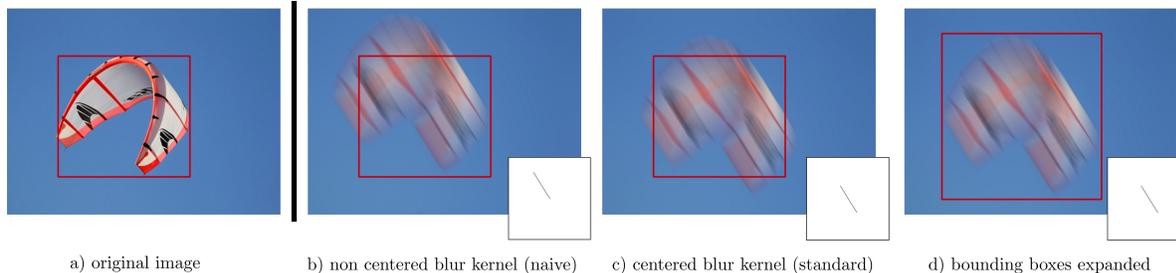
Figure 4. a) An image from the MS COCO [26] trainset with an associated bounding box label. b) The same image but blurred and with the same non-translated bounding box, now introducing a training/evaluation mismatch. The object may have been there at the start of the exposure (or end if the kernel smeared the other way), but this is certainly not true for the rest of the image. c) The first remedy, centering the kernel by shifting the barycenter of the nonzero points to the center of the filter. d) The object is smeared outside of the original COCO bounding box, so the label is expanded using the filter's max/min points to capture the superset of object locations.

in accuracy can be up to 8-10mAP@50 (see "Non-Centered Augmented" and "Standard Augmentated" in Fig. 5) points with the most severe blur. All networks shown here will be trained and evaluated with centered kernels, except when explicitly mentioned. We include ablation experiments with no-centering in the supplemental.

Expanding Target Boxes Compared to the original bounding box, the expanded label can cover the superset of pixels where an object projected during an exposure; see Fig. 4(d). A worst-case scenario could occur without this correction during training: for a small object, the sharp image's label could seemingly miss the blurred object entirely due to IOU cutoffs. As a remedy, for every generated centered kernel, we find the maximum offsets for non zero kernel elements in both 2D axes, x^-, x^+, y^-, y^+, and use them to expand the boundaries of COCO bounding box labels. The new bounding box labels (top left and width/height) are now

$$
\begin{aligned}
\hat{b}_x &= b_x - |x^-| \\
\hat{b}_y &= b_y - |y^-| \\
\hat{b}_w &= b_w + |x^-| + |x^+| \\
\hat{b}_h &= b_h + |y^-| + |y^+|.
\end{aligned}
\tag{5}
$$

We train variants of our networks with these expanded boxes alongside kernel centering. During test time, we evaluate these networks using expanded bounding boxes.

4.5. Specializing for Categories of Blur

The final category of remedies explores if egomotion induced blur is perhaps multiple problems masquerading as one. We explore training blur specialized networks on specific partitioned segments of the blur space, as if categories of blur are mutltiple distinct tasks. The findings on recognition in [51] show that specialized networks can *sometimes* achieve higher task accuracy on their respective blur types than general blur augmented networks.

Two Specialized Meta-Models We make two sets of specialized networks, that differ in how the motion kernels are clustered into categories. First, motion blur is grouped based on the type of kernel P, alone, leading to a bespoke network for each of P_{1-3} with a fourth generalist network trained for all types and exposures.

The second grouping creates three networks specialized at each P but exclusively on long exposure blur. One further network handles all low exposure blur. As per [12], networks are biased toward texture. Instead of using this knowledge to create more corruption robust networks, it is exploited here to make more shape biased networks for substantial motion blur.

Blur Estimation and Network Selection A ResNet-18 blur estimator module is added, and runs $10\times$ faster than the detection framework. The estimator categorizes the blur present in the image at test time. One network is trained on 16 classes (sharp and the combinations of all exposures and blur types) and the other network focuses on the separation between specific blur types at high exposures and general blur at low exposure (four classes). Details of how these estimators are trained are in the supplemental materiel.

5. Comparisons and Evaluation

We report COCO minival results for all proposed remedies, at both test time and train time. Detection accuracy at mAP@50 is reported for all below-listed models and variants in Fig. 5 and Fig. 6, where the former uses COCO original labels, and the latter uses expanded labels. We also report accuracy results on two pseudo-real blur datasets, GO-PRO [31] and REDS [30], and a real-world blur dataset, RealBlur [40], obtained using shutter tied cameras. These datasets don't have box annotations, so we utilize a state-of-the-art high accuracy detector, DetectoRS [35], to obtain pseudo-groundtruth bounding-boxes for evaluation. For evaluating expanded bounding boxes, we generate our own GOPRO testset using grountruth sharp frames and use flow
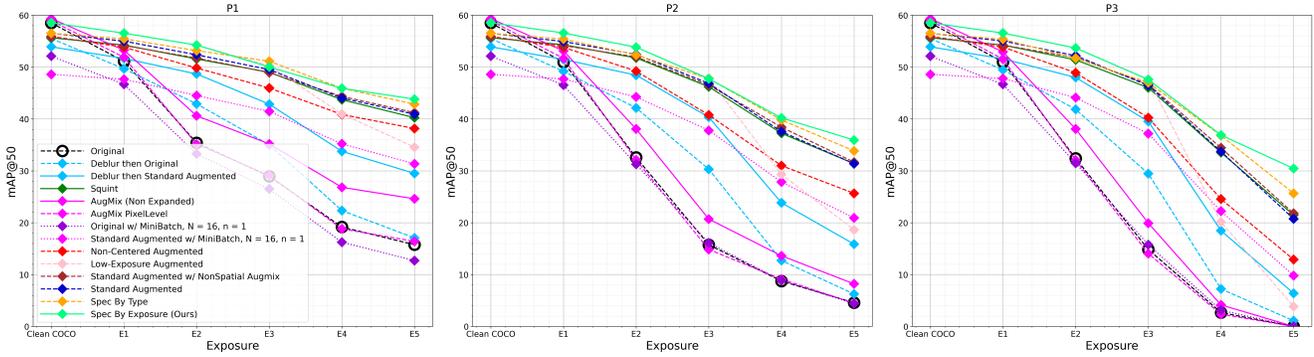
Figure 5. Non-expanded Standard Labeling COCO minival mAP@50 accuracy values across the three blur types and then exposures. Almost all methods and hybrids improve beyond the Original network. There is no benefit in augmenting for blur, and then using either minibatch statistics or a deblurer network.
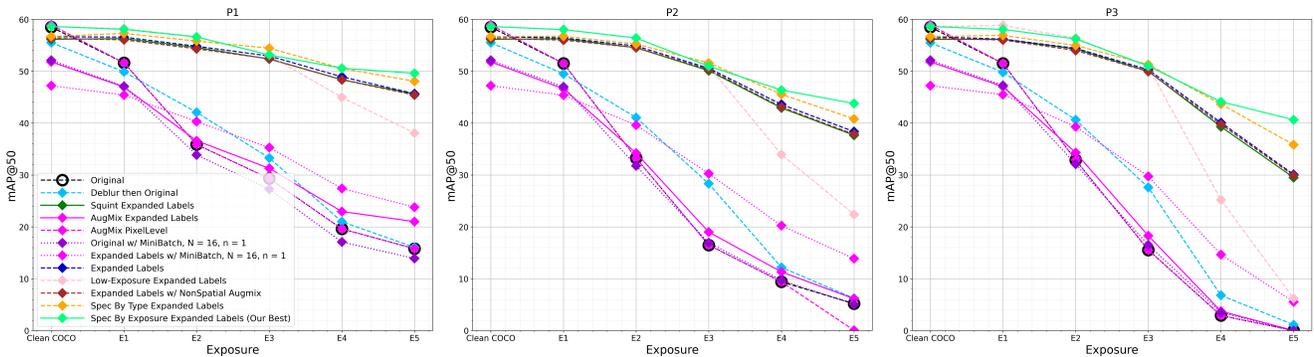

Figure 6. Expanded Labeling COCO minival mAP@50 accuracy values across the three blur types and then exposures. The Spec by Exposure network ("Ours") excels at both ends of the exposure extremes, likely due to the biased training and specialized networks it enjoys. All expanded-box trained networks (except AugMix trained on expanded labels via spatial augmentation) perform better than their standard counterparts. Networks augmented with both blur and non-spatial AugMix perform well at low exposures and dataset generalization; in Table 1 we show performance for Spec By Exposure that makes use of this for improved performance on other datasets.

computed using [50] for bounding-box expansion.

Names in the figures are explained below, and map to the five remedy categories. Qualitative video here: visual. cs.ucl.ac.uk/pubs/handlingMotionBlur.

- **Standard Augmented** and **Expanded Labels** were trained on non-expanded but centered COCO labels, and expanded and centered COCO labels respectively. Both were trained on a 10/90 mixture of sharp to blurry images across all blur types.

- **Deblur then original** and **Deblur then Standard Augmented** are both modes of operation where the image is first deblurred using [31] then run through either the original network or a **Standard Augmented** network respectively.

- **Squint** and **Squint Expanded Labels** come from Sec. 4.2, and have been trained either using standard labels under blur or expanded labels, respectively.

- **AugMix [16]** Hendrycks *et al.* [16] As described in

Sec. 4.3, we evaluate a non spatial version, **AugMix PixelLevel**, a spatial version without label expansion, **AugMix** [16], and a version trained with expanded labels as per augmentations and evaluated with blur based label expansions **AugMix Expanded Labels**.

- **Standard Augmented w/MiniBatch** and **Expanded Labels w/MiniBatch** follow Schneider *et al.* [43] and use modified minibatch normalization with $N = 16$ and $n = 1$ as in Sec. 4.3 with networks that have been augmented for blur using either standard labels or expanded labels respectively.

- **Standard Augmented w/ NonSpatial AugMix** and **Expanded Labels w/ NonSpatial AugMix** have been trained by first transforming the image using non spatial AugMix then blurring the image and training with expanded labels. NonSpatial AugMix augmentation helps when generalizing to other datasets at low exposure blur.

Model	GOPRO [31] Sharp	GOPRO [31] Blurry	RealBlur [20] Sharp	RealBlur [20] Blurry	REDS [30] Sharp	REDS [30] Blurry	GOPRO Expanded* Sharp	GOPRO Expanded* Blurry
Spec by Exposure w/ †	34.89	**28.15**	43.23	<u>36.37</u>	39.86	**32.55**	38.06	30.99
Low-Exp Net w/ NonSpatial AugMix †	34.89	<u>27.87</u>	43.23	36.28	39.86	<u>32.53</u>	38.06	30.92
Spec by Exposure	34.80	27.05	42.91	35.26	40.06	31.51	36.79	<u>31.14</u>
Spec by Type	32.42	27.50	40.91	36.11	36.96	30.33	35.26	31.01
Low-Exposure Net	34.80	26.91	42.91	35.20	40.06	31.47	36.79	**31.60**
Standard Aug w/ NonSpatial AugMix‡	32.16	27.05	42.77	**36.82**	35.61	28.72	35.18	29.94
Standard Augmented	32.42	26.54	40.93	35.63	36.96	29.24	35.26	30.75
Non-Centered Augmented	33.08	25.66	40.47	34.47	36.82	28.80	36.74	28.47
Standard Aug w/ Minibatch	30.06	23.78	35.01	31.32	28.99	24.94	33.32	26.79
Original w/ NonSpatial AugMix	33.68	19.10	43.90	29.63	43.38	23.52	36.63	22.16
Original w/ Minibatch	31.78	16.73	38.54	27.17	36.03	22.18	33.45	20.50
Deblur then Standard Augmented	12.53	5.14	32.53	28.90	34.53	28.42	34.54	29.72
Deblur then Original	10.58	2.24	31.45	28.76	40.56	31.84	35.74	26.70
Original	35.85	19.64	42.39	29.00	42.02	24.38	36.33	22.30

Table 1. mAP@0.5 for models trained on COCO images with synthesized blur and evaluated on real world blur datasets using predictions on sharp images using DetectoRS [35] as groundtruth. While NonSpatial AugMix doesn't improve performance for blur augmentation on the COCO minival, it does increase the performance of the COCO trained Low-Exposure† and Standard Augmented‡ networks on real world datasets. Spec by Exposure w/† utilizes the NonSpatial AugMix augmentation version of the Low-Exposure network. *Expanded augmented trained versions of the networks are used on this expanded labels test-set.

- **Spec by Type** is the first bag of specialists from Sec. 4.5 where the **Standard Augment** network is used when no blur is detected. **Spec by Exposure** is the second bag of specialists where each of three networks specializes in one P, but only at long exposures, and one network handles all short exposures and sharp images, **Low-Exposure Augment**. Again a blur estimator trained for these classes selects the right network. **Spec by Type Expanded Labels** and **Spec by Exposure Expanded Labels** are obviously variants trained on expanded labels. These networks and associated mode of operation outperform the rest due to their exploiting of the texture *vs*. bias trade-off and the use of an accurate blur estimator. Notably, the network responsible for sharp and low exposure blur recovers the accuracy lost on sharp images usually associated with blur augmentation networks.

6. Discussion

We achieve state-of-the-art object detection results for egomotion-blurred images. We have succeeded in identifying that two factors adversely affect detection in such images. The first is that labels for sharp images should be customized for the motion-blur domain. In our remedies, that means translating and expanding the bounding box labels to match the blurred versions of relevant objects. The second is that categories of motion blur are distinct enough for the model to be trained for each blur-category separately. Interestingly, the second factor is the opposite of what [51] found with recognition tasks, where mixing blur-types during training was effective.

Through our "differential diagnosis" approach, the other three factors explored here seem unpromising for explaining the destructiveness of blur on CNN-based detection. These negative results are not conclusive, as the remedies may simply be immature. For example, better deblurring may eventually restore missing texture at all scales.

In the future, to reduce the memory footprint of our favored solution, the blur-selector and distinct exposure-specific models could be combined into one multi-task model. They are already end-to-end differentiable, but then they could share layers. Further progress in this direction could benefit from a distilled dataset that allows for detection labels and blur from real data, perhaps through the use of event driven cameras or multi-camera datasets.

One clear limitation is that even sharp images have < 60 $mAP@0.5$ detection accuracy with a realtime-capable backbone, before blur hurts the situation further. Depth or disparity data would help address scenes with dynamic blur, since the blur kernels are depth-dependent. 360-cameras, augmented as proposed here for COCO, could be beneficial for dealing with sharp or blurred target objects that are partially outside a typical camera's field of view. The impact of our approach could be especially helpful in particular applications, such as drone-based following, where even brief interruptions in tracking can ruin a film.

References

[1] Miika Aittala and Frédo Durand. Burst image deblurring using permutation invariant convolutional neural networks. In *Proceedings of the European Conference on Computer Vision (ECCV)*, pages 731–747, 2018. 3

[2] Gedas Bertasius, Lorenzo Torresani, and Jianbo Shi. Object detection in video with spatiotemporal sampling networks. In *Proceedings of the European Conference on Computer Vision (ECCV)*, September 2018. 1

[3] Giacomo Boracchi and Alessandro Foi. Modeling the performance of image restoration from motion blur. *IEEE Transactions on Image Processing*, 21(8):3502–3517, 2012. 3, 4

[4] Tim Brooks and Jonathan T Barron. Learning to synthesize motion blur. In *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition*, pages 6840–6848, 2019. 4

[5] Ayan Chakrabarti. A neural approach to blind motion deblurring. In *European conference on computer vision*, pages 221–235. Springer, 2016. 3

[6] K. Chatfield, V. Lempitsky, A. Vedaldi, and A. Zisserman. The devil is in the details: an evaluation of recent feature encoding methods. In *British Machine Vision Conference*, 2011. 1

[7] K. Chatfield, K. Simonyan, A. Vedaldi, and A. Zisserman. Return of the devil in the details: Delving deep into convolutional nets. In *British Machine Vision Conference*, 2014. 1

[8] Liang Chen, Faming Fang, Tingting Wang, and Guixu Zhang. Blind image deblurring with local maximum gradient prior. In *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition*, pages 1742–1750, 2019. 2

[9] Ekin D Cubuk, Barret Zoph, Dandelion Mane, Vijay Vasudevan, and Quoc V Le. Autoaugment: Learning augmentation strategies from data. In *Proceedings of the IEEE conference on computer vision and pattern recognition*, pages 113–123, 2019. 3

[10] J. Deng, W. Dong, R. Socher, L.-J. Li, K. Li, and L. Fei-Fei. ImageNet: A Large-Scale Hierarchical Image Database. In *CVPR09*, 2009. 3

[11] Rob Fergus, Barun Singh, Aaron Hertzmann, Sam T Roweis, and William T Freeman. Removing camera shake from a single photograph. In *ACM SIGGRAPH 2006 Papers*, pages 787–794. 2006. 2

[12] Robert Geirhos, Patricia Rubisch, Claudio Michaelis, Matthias Bethge, Felix A Wichmann, and Wieland Brendel. Imagenet-trained cnns are biased towards texture; increasing shape bias improves accuracy and robustness. *arXiv preprint arXiv:1811.12231*, 2018. 6

[13] Kaiming He, Xiangyu Zhang, Shaoqing Ren, and Jian Sun. Deep residual learning for image recognition. In *Proceedings of the IEEE conference on computer vision and pattern recognition*, pages 770–778, 2016. 4

[14] Dan Hendrycks, Steven Basart, Norman Mu, Saurav Kadavath, Frank Wang, Evan Dorundo, Rahul Desai, Tyler Zhu, Samyak Parajuli, Mike Guo, et al. The many faces of robustness: A critical analysis of out-of-distribution generalization. *arXiv preprint arXiv:2006.16241*, 2020. 3

[15] Dan Hendrycks and Thomas Dietterich. Benchmarking neural network robustness to common corruptions and perturbations. *arXiv preprint arXiv:1903.12261*, 2019. 3, 4, 5

[16] Dan Hendrycks, Norman Mu, Ekin D Cubuk, Barret Zoph, Justin Gilmer, and Balaji Lakshminarayanan. Augmix: A simple data processing method to improve robustness and uncertainty. *arXiv preprint arXiv:1912.02781*, 2019. 2, 3, 5, 7

[17] Dan Hendrycks, Kevin Zhao, Steven Basart, Jacob Steinhardt, and Dawn Song. Natural adversarial examples. *arXiv preprint arXiv:1907.07174*, 2019. 3

[18] Tae Hyun Kim, Byeongjoo Ahn, and Kyoung Mu Lee. Dynamic scene deblurring. In *Proceedings of the IEEE International Conference on Computer Vision*, pages 3160–3167, 2013. 2

[19] Max Jaderberg, Karen Simonyan, Andrew Zisserman, et al. Spatial transformer networks. In *Advances in neural information processing systems*, pages 2017–2025, 2015. 2, 5

[20] Jucheol Won Sunghyun Cho Jaesung Rim, Haeyun Lee. Real-world blur dataset for learning and benchmarking deblurring algorithms. In *Proceedings of the European Conference on Computer Vision (ECCV)*. 8

[21] Licheng Jiao, Fan Zhang, Fang Liu, Shuyuan Yang, Lingling Li, Zhixi Feng, and Rong Qu. A survey of deep learning-based object detection. *IEEE Access*, 7:128837–128868, 2019. 4

[22] Neel Joshi, Sing Bing Kang, C Lawrence Zitnick, and Richard Szeliski. Image deblurring using inertial measurement sensors. *ACM Transactions on Graphics (TOG)*, 29(4):1–9, 2010. 3

[23] BR Kapuriya, Debasish Pradhan, and Reena Sharma. Detection and restoration of multi-directional motion blurred objects. *Signal, Image and Video Processing*, 13(5):1001–1010, 2019. 2

[24] Orest Kupyn, Volodymyr Budzan, Mykola Mykhailych, Dmytro Mishkin, and Jiří Matas. Deblurgan: Blind motion deblurring using conditional adversarial networks. In *Proceedings of the IEEE conference on computer vision and pattern recognition*, pages 8183–8192, 2018. 3

[25] Tsung-Yi Lin, Piotr Dollár, Ross Girshick, Kaiming He, Bharath Hariharan, and Serge Belongie. Feature pyramid networks for object detection. In *Proceedings of the IEEE conference on computer vision and pattern recognition*, pages 2117–2125, 2017. 4

[26] Tsung-Yi Lin, Michael Maire, Serge Belongie, James Hays, Pietro Perona, Deva Ramanan, Piotr Dollár, and C Lawrence Zitnick. Microsoft coco: Common objects in context. In *European conference on computer vision*, pages 740–755. Springer, 2014. 1, 3, 4, 6

[27] Li Liu, Wanli Ouyang, Xiaogang Wang, Paul Fieguth, Jie Chen, Xinwang Liu, and Matti Pietikäinen. Deep learning for generic object detection: A survey. *International journal of computer vision*, 128(2):261–318, 2020. 4

[28] Leon B Lucy. An iterative technique for the rectification of observed distributions. *The astronomical journal*, 79:745, 1974. 2

[29] Claudio Michaelis, Benjamin Mitzkus, Robert Geirhos, Evgenia Rusak, Oliver Bringmann, Alexander S Ecker,

Matthias Bethge, and Wieland Brendel. Benchmarking robustness in object detection: Autonomous driving when winter is coming. *arXiv preprint arXiv:1907.07484*, 2019. 3, 4, 5

[30] 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 *The IEEE Conference on Computer Vision and Pattern Recognition (CVPR) Workshops*, June 2019. 6, 8

[31] Seungjun Nah, Tae Hyun Kim, and Kyoung Mu Lee. Deep multi-scale convolutional neural network for dynamic scene deblurring. In *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition*, pages 3883–3891, 2017. 2, 3, 4, 5, 6, 7, 8

[32] Jinshan Pan, Deqing Sun, Hanspeter Pfister, and Ming-Hsuan Yang. Blind image deblurring using dark channel prior. In *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition*, pages 1628–1636, 2016. 2

[33] Liyuan Pan, Cedric Scheerlinck, Xin Yu, Richard Hartley, Miaomiao Liu, and Yuchao Dai. Bringing a blurry frame alive at high frame-rate with an event camera. In *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition*, pages 6820–6829, 2019. 3

[34] Adam Paszke, Sam Gross, Francisco Massa, Adam Lerer, James Bradbury, Gregory Chanan, Trevor Killeen, Zeming Lin, Natalia Gimelshein, Luca Antiga, Alban Desmaison, Andreas Kopf, Edward Yang, Zachary DeVito, Martin Raison, Alykhan Tejani, Sasank Chilamkurthy, Benoit Steiner, Lu Fang, Junjie Bai, and Soumith Chintala. Pytorch: An imperative style, high-performance deep learning library. In H. Wallach, H. Larochelle, A. Beygelzimer, F. dAlché-Buc, E. Fox, and R. Garnett, editors, *Advances in Neural Information Processing Systems 32*, pages 8024–8035. Curran Associates, Inc., 2019. 4

[35] Siyuan Qiao, Liang-Chieh Chen, and Alan Yuille. Detectors: Detecting objects with recursive feature pyramid and switchable atrous convolution. *arXiv preprint arXiv:2006.02334*, 2020. 6, 8

[36] Ramesh Raskar, Amit Agrawal, and Jack Tumblin. Coded exposure photography: motion deblurring using fluttered shutter. In *ACM SIGGRAPH 2006 Papers*, pages 795–804. 2006. 3

[37] Adria Recasens, Petr Kellnhofer, Simon Stent, Wojciech Matusik, and Antonio Torralba. Learning to zoom: a saliency-based sampling layer for neural networks. In *Proceedings of the European Conference on Computer Vision (ECCV)*, pages 51–66, 2018. 2, 5

[38] Shaoqing Ren, Kaiming He, Ross Girshick, and Jian Sun. Faster r-cnn: Towards real-time object detection with region proposal networks. In *Advances in neural information processing systems*, pages 91–99, 2015. 3

[39] William Hadley Richardson. Bayesian-based iterative method of image restoration. *JoSA*, 62(1):55–59, 1972. 2

[40] Jaesung Rim et al. *Real-World Blur Dataset for Learning and Benchmarking Deblurring Algorithms*. PhD thesis, DGIST, 2020. 4, 5, 6

[41] Evgenia Rusak, Lukas Schott, Roland Zimmermann, Julian Bitterwolf, Oliver Bringmann, Matthias Bethge, and Wieland Brendel. Increasing the robustness of dnns against image corruptions by playing the game of noise. *arXiv preprint arXiv:2001.06057*, 2020. 3

[42] Uwe Schmidt, Carsten Rother, Sebastian Nowozin, Jeremy Jancsary, and Stefan Roth. Discriminative non-blind deblurring. In *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition*, pages 604–611, 2013. 2, 4

[43] Steffen Schneider, Evgenia Rusak, Luisa Eck, Oliver Bringmann, Wieland Brendel, and Matthias Bethge. Improving robustness against common corruptions by covariate shift adaptation. *Advances in Neural Information Processing Systems*, 33, 2020. 2, 3, 5, 7

[44] Christian J Schuler, Michael Hirsch, Stefan Harmeling, and Bernhard Schölkopf. Learning to deblur. *IEEE transactions on pattern analysis and machine intelligence*, 38(7):1439–1451, 2015. 2, 3

[45] Shuochen Su, Mauricio Delbracio, Jue Wang, Guillermo Sapiro, Wolfgang Heidrich, and Oliver Wang. Deep video deblurring for hand-held cameras. In *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition*, pages 1279–1288, 2017. 2

[46] Jian Sun, Wenfei Cao, Zongben Xu, and Jean Ponce. Learning a convolutional neural network for non-uniform motion blur removal. In *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition*, pages 769–777, 2015. 3

[47] Yu-Wing Tai, Hao Du, Michael S Brown, and Stephen Lin. Image/video deblurring using a hybrid camera. In *2008 IEEE Conference on Computer Vision and Pattern Recognition*, pages 1–8. IEEE, 2008. 3

[48] Chang Tang, Xinzhong Zhu, Xinwang Liu, Lizhe Wang, and Albert Zomaya. Defusionnet: Defocus blur detection via recurrently fusing and refining multi-scale deep features. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR)*, June 2019. 2

[49] Xin Tao, Hongyun Gao, Xiaoyong Shen, Jue Wang, and Jiaya Jia. Scale-recurrent network for deep image deblurring. In *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition*, pages 8174–8182, 2018. 2, 3

[50] Zachary Teed and Jia Deng. Raft: Recurrent all-pairs field transforms for optical flow. In *European Conference on Computer Vision*, pages 402–419. Springer, 2020. 7

[51] Igor Vasiljevic, Ayan Chakrabarti, and Gregory Shakhnarovich. Examining the impact of blur on recognition by convolutional networks. *arXiv preprint arXiv:1611.05760*, 2016. 3, 4, 5, 6, 8

[52] Oliver Whyte, Josef Sivic, Andrew Zisserman, and Jean Ponce. Non-uniform deblurring for shaken images. *International journal of computer vision*, 98(2):168–186, 2012. 2

[53] Li Xu, Shicheng Zheng, and Jiaya Jia. Unnatural l0 sparse representation for natural image deblurring. In *Proceedings of the IEEE conference on computer vision and pattern recognition*, pages 1107–1114, 2013. 2

[54] Xiangyu Xu, Jinshan Pan, Yu-Jin Zhang, and Ming-Hsuan Yang. Motion blur kernel estimation via deep learning. *IEEE Transactions on Image Processing*, 27(1):194–205, 2017. 3

[55] J. Zhang, J. Pan, J. Ren, Y. Song, L. Bao, R. W. H. Lau, and M. Yang. Dynamic scene deblurring using spatially variant recurrent neural networks. In *2018 IEEE/CVF Conference on Computer Vision and Pattern Recognition*, pages 2521–2529, 2018. 3

[56] Kaihao Zhang, Wenhan Luo, Yiran Zhong, Lin Ma, Bjorn Stenger, Wei Liu, and Hongdong Li. Deblurring by realistic blurring. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, pages 2737–2746, 2020. 4

[57] Shaojie Zhuo, Dong Guo, and Terence Sim. Robust flash deblurring. In *2010 IEEE Computer Society Conference on Computer Vision and Pattern Recognition*, pages 2440–2447. IEEE, 2010. 3