Task-Driven Super Resolution: Object Detection in Low-Resolution Images

Muhammad Haris[1(✉)], Greg Shakhnarovich[2], and Norimichi Ukita[3]

[1] Universitas Nusa Mandiri, Jakarta, Indonesia
muhammad.uhs@nusamandiri.ac.id
[2] Toyota Technological Institute at Chicago, Chicago, USA
greg@ttic.edu
[3] Toyota Technological Institute, Nagoya, Japan
ukita@toyota-ti.ac.jp

Abstract. We consider how image super-resolution (SR) can contribute to an object detection task in low-resolution images. Intuitively, SR gives a positive impact on the object detection task. While several previous works demonstrated that this intuition is correct, SR and detector are optimized independently in these works. This paper analyze a framework to train a deep neural network where the SR sub-network explicitly incorporates a detection loss in its training objective, via a tradeoff with a traditional detection loss. This end-to-end training procedure allows us to train SR preprocessing for any differentiable detector. We demonstrate extensive experiments that show our task-driven SR consistently and significantly improves the accuracy of an object detector on low-resolution images from COCO and PASCAL VOC data set for a variety of conditions and scaling factors.

Keywords: Super-resolution · Object detection · End-to-end learning · Task network · Machine perception · Joint optimization

1 Introduction

Image Super-Resolution (SR) belongs to image restoration and enhancement (e.g., denoising and deblurring) algorithms, widely studied in computer vision and graphics. In both communities, the goal is to reconstruct an image from a degenerated version as accurately as possible. The quality of the reconstructed image is evaluated by pixel-based quantitative metrics such as PSNR (peak signal-to-noise ratio) and SSIM (structure similarity) [15]. Recently-proposed perceptual quality ([2,14]) can also be employed for evaluation as well as for optimizing the reconstruction model. Relationships between the pixel-based and perceptual quality metrics have been investigated in the literature ([4,9]) in order to harmonize these two kinds of metrics. Ultimately, the goal of SR is still to restore an image as well as possible in accordance with criteria in human visual perception.

Main work has been done during postdoctoral at TTI Japan.

T. Mantoro et al. (Eds.): ICONIP 2021, CCIS 1516, pp. 387–395, 2021.
https://doi.org/10.1007/978-3-030-92307-5_45

(a) HR (b) LR (c) Bicubic SR PSNR: 21.26 (d) SR (no task) PSNR: 22.02 (e) TDSR (proposed) PSNR: 21.54

Fig. 1. Scale sensitivity in object detection and the effectiveness of our proposed method (i.e., end-to-end learning in accordance with the mutual improvement of SR and object detection tasks). Images shown in the top row show (a) an original high resolution image, (b) its low-resolution image (here $1/8$-size, padded with black), (c) SR image obtained by bicubic interpolation, (d) SR image obtained by the SR model optimized with no regard to detection, and (e) SR image obtained by our proposed task-driven SR method, using the same model as in (d). For each of the reconstructed HR images, we also report PSNR w.r.t. the original. Despite ostensibly lower PSNR, the TDSR result recovers the correct detection results with high scores, in this case even suppressing a false detection present in the original HR input.

We propose to bridge this isolation by explicitly incorporating the objective of the downstream task (such as object detection) into training of an SR module. Figure 1 illustrates the effect of our proposed, task-driven approach to SR. Our proposal (e) generated from a low-resolution (LR) image (b) can successfully bring recognition accuracy close to the score of their original high-resolution (HR) image (a).

Our approach is motivated by two observations. (1) SR is ill-posed. Many possible HR images when downsampled produce the same LR image. We expect that the additional cue given by the downstream task objective such as detection may help guide the SR solution. (2) Human perception and machine perception differ. It is known that big differences are observed between human and machine perceptions, in particular, with highly-complex deep networks. Thus, if our goal is to super-resolve an image in part for machine perception, we believe it is prudent to explicitly "cater" to the machine perception when learning SR.

The main contributions of this paper are:

- An approach to SR that uses the power of end-to-end training in deep learning to combine low-level and high-level vision objectives, leading to what we call *Task-Driven Super Resolution* (TDSR). As a means of increasing robustness of SR methods for computer vision tasks, this approach provides results substantially better than other SR methods, and is potentially applicable to a broad range of low-level image processing tools and high-level tasks.
- A simple yet effective view of SR, explicitly acknowledging the generative or semantic aspects of SR in high scaling factors, which we hope will encourage additional work in the community to help further reduce the gap between low-level and high-level vision.
- Extensive experiments to handle more difficult scenarios where the image are afflicted by additional sources of corruption such as blur and noise.

2 Related Work

2.1 Image Super Resolution

A huge variety of image SR techniques have been proposed; see survey papers ([13,16]) for more details.

Like other vision problems, SR has benefited from recent advances in deep convolutional neural networks (DCNNs). One of most notable work is DBPN-SR ([5]). It shares the SR features at different scales by iterative forward and backward projections and enables the networks to preserve the HR components by learning various up- and down-sampling operators while generating deeper features. While deep features provided by DCNNs allow us to preserve clear high-frequency photo-realistic textures, it is difficult to completely eliminate blur artifacts. This problem has been addressed by introduction of novel objectives, such as perceptual similarity ([2,7]) and adversarial losses ([3,17]). Finally, the two ideas can be combined, incorporating perceptual similarity into generative adversarial networks (GANs) in SRGAN ([10]). In contrast to prior work, we explicitly incorporate the objective of a well defined, discriminative task (such as detection) into the SR framework.

2.2 Detection of Small Objects

One of the remaining problems in computer vision, such as object detection and scene parsing, is to detect small objects. This issue has been investigated by ([6,8]). Most of these methods proposed context-aware network by re-scaling the input to several resolutions then training the networks at each resolution or proposing a mechanism to select the pooling field size to preserve the small details. Here we consider an alternative: transform the LR images into HR images using SR. So that, instead of designing more LR friendly detector, we can try to make LR images "look like HR image", for which we have plenty of examples, in the hope that the existing detector "used to HR" will then be able to detect objects. In other words, rather than improve the detector, we pre-process the input to make it more amenable to the detector as is.

3 Task Driven Super-Resolution

Our method relies on two building blocks: an SR network S and a task network D as shown in Fig. 2. The SR network maps an LR image x^l to an HR image x^h producing an SR image $x^{sr} = S(x^l; \theta_{SR})$, where θ_{SR} denotes all the parameters of the SR network. The task network takes an image x^{sr} and outputs a (possibly structured) prediction $\widehat{y} = D(x^{sr}; \theta_D)$. We refer to these predictors as "networks" because they are likely to be deep neural networks. However our approach does not presume anything about S and D beyond differentiability for training the whole network with an end-to-end learning scheme.

We assume that the task network D has been trained and its parameters θ_D remain fixed throughout training (and will, for brevity, be omitted from notation).

Our method is applicable to any task network. It can be used for a variety of tasks, for example, depth estimation or semantic segmentation. However, in this paper, we

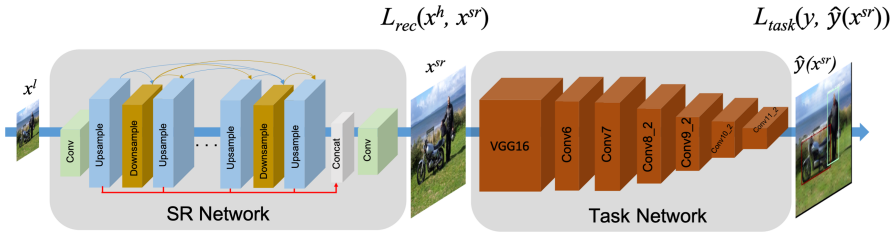


Fig. 2. Network Architecture. Here, we use DBPN ([5]) as an SR network and SSD ([11]) as a task network concatenate to perform end-to-end training.

restrict our attention to the object detection task, in which $\widehat{y}$ consists of a set of scored bounding boxes for given object classes.

3.1 Component Networks

We use the recently proposed Deep Back-Projection Networks (DBPN) [5] as the SR component. The DBPN achieve state of the art or competitive results on standard SR benchmarks, when trained with the MSE reconstruction loss

$$L_{rec}\left(x^h, x^{sr}\right) = \frac{1}{N}\sum_{i=1}^{N}(x_i^h - x_i^{sr})^2 \tag{1}$$

where i ranges of the N pixel indices in the HR image x^h.

As the detector, we use the Single Shot MultiBox Detector (SSD) [11]. The SSD detector works with a set of default bounding boxes, covering a range of positions, scales and aspect ratios; each box is scored for presence of an object from every class. Given the ground truth for an image x^h, B is the number of matched default boxes to the ground truth boxes y. These matched boxes form the predicted detections $\widehat{y}(x^{sr})$. The task (detection) loss of SSD is combined of confidence loss and localization loss:

$$L_{task}(y, \widehat{y}(x^{sr})) = \frac{1}{B}\left[L_{conf}(y, \widehat{y}(x^{sr})) + \lambda L_{loc}(y, \widehat{y}(x^{sr}))\right] \tag{2}$$

The confidence loss L_{conf} penalizes incorrect class predictions for the matched boxes. The localization loss L_{loc} penalizes displacement of boxes vs. the ground truth, using smooth L_1 distance. Both losses in (2) are differentiable with respect to their inputs.

Importantly, every default bounding box in SSD is associated with a set of cells in feature maps (activation layers) computed by a convolutional neural network. As a result, since the loss in (2) decomposes over boxes, it is a differentiable function of the network activations and thus a function of the pixels in the input image, allowing us to incorporate this task loss in the TDSR objective described below.

3.2 Task Driven Training

Normally, learning-based SR systems are trained using some sort of reconstruction loss L_{rec}, such as mean (over pixels) squared error (MSE) between x^h and x^{sr}. In contrast,

the detector is trained with L_{task} intended to improve the measure of its accuracy, typically measured as the average precision (AP) for one class, and the mean AP (mAP) over classes for the entire data set.

Let x^h be the image with detection ground truth labels y, and x^l is a downscaled image by a fixed factor. We propose the compound loss, which on the example (x^h, y) is given by

$$\begin{aligned} L(x^h, y; \theta_{SR}) = & \alpha L_{rec}\left(x^h, S(x^l; \theta_{SR})\right) + \\ & \beta L_{task}\left(y, D(S(x^l; \theta_{SR}); \theta_D)\right) \end{aligned} \tag{3}$$

where α and β are weights determining relative strength of the reconstruction loss and the detection loss. Under the assumption that both S and D are differentiable, we can use the chain rule, and compute the gradient of L_{task} with respect to its input, the super-resolved x^l. Then this per-pixel gradient is combined with the per-pixel gradient of the reconstruction loss L_{rec}. The SR parameters θ_{SR} are then updated using standard back-propagation from this combined gradient:

$$\begin{aligned} & \alpha \frac{\partial}{\partial \theta_{SR}} L_{rec}\left(x^h, S(x^l; \theta_{SR})\right) + \\ & \beta \frac{\partial L_{task}\left(y, D(S(x^l); \theta_D)\right)}{\partial S(x^l)} \frac{\partial S(x^l)}{\partial \theta_{SR}} \end{aligned} \tag{4}$$

4 Experimental Results

4.1 Implementation Details

Base networks. DBPN ([5]) constructs mutually-connected up- and down-sampling layers each of which represents different types of image degradation and HR components. The stack of up- and down- projection units creates an efficient way to iteratively minimize the reconstruction error, to reconstruct a huge variety of SR features, and to enable large scaling factors such as $8\times$ enlargement. We used the setting recommended by the authors: "a 8×8 convolutional layer with four striding and two padding" and "a 12×12 convolutional layer with eight striding and two padding" are used for $4\times$ and $8\times$ SRs, respectively, in order to construct a projection unit. Here, we use D-DBPN which is one of DBPN variants. For object detection, we use SSD300 where the input size is 300×300 pixels. The network uses VGG16 through `conv5_3` layer, then uses `conv4_3`, `conv7` (`fc7`), `conv8_2`, `conv9_2`, `conv10_2`, and `conv11_2` as feature maps to predict the location and confidence score of each detected object. The code for both networks are publicly accessible in the internet.

Datasets. We initialized all experiments with DBPN model pretrained on the DIV2K data set ([1]), made available by the authors of ([5]). We used SSD network pretrained on PASCAL VOC0712 `trainval` and MSCOCO `train2017`. When fine-tuning DBPN in our experiments, with or without task-driven objective, we reused PASCAL VOC0712 `trainval` and MSCOCO `train2017`, with data augmentation. The augmentation consists of photometric distortion, scaling, flipping, random cropping that are recommended to train SSD. Test images on VOC2007 `test` and MSCOCO `val2017`

Table 1. VOC2007 `test` detection results on $4\times$ and $8\times$.

Scale	Method	n-iter : w_{td}	PSNR	AP	aero	bike	bird	boat	bottle	bus	car	cat	chair	cow	table	dog	horse	bike	person	plant	sheep	sofa	train	tv
	HR	-	-	75.8	79.3	85.4	74.1	68.9	46.6	83.7	85.5	86.1	59.1	81.3	77.1	83.5	85.2	82.9	77.6	46.7	73.8	79.9	84.8	73.8
	LR	-	-	41.7	48.9	46.8	33.5	31.9	10.7	57.7	48.6	55.9	18.5	31.7	50.1	50.2	61.3	54.2	45.0	18.5	32.8	52.3	52.9	33.4
	Bicubic	-	25.30	41.3	50.9	43.9	37.3	22.0	14.5	53.2	53.9	55.8	18.8	35.6	37.9	52.1	56.9	53.5	49.5	18.7	40.3	51.1	41.8	38.5
	SRGAN	-	23.51	44.6	62.2	45.0	37.0	29.3	15.9	63.0	56.7	44.6	26.5	40.4	46.4	47.9	59.2	52.1	53.1	18.1	40.5	56.9	48.6	47.9
$4\times$	DBPN	-	22.87	41.9	61.3	41.5	34.4	25.4	16.1	57.7	55.1	43.4	28.9	35.6	44.2	40.7	52.4	47.3	50.0	15.6	32.5	59.1	47.0	50.2
	SR-FT	$100k : 0$	26.65	52.6	59.5	61.7	44.3	33.5	26.5	65.6	63.8	61.2	36.2	45.1	55.5	55.7	67.6	64.3	59.4	21.8	45.3	65.8	58.6	60.2
	SR-FT+	$100k : 1 : 0 + 200k : 1 : 0$	26.72	53.6	59.6	62.9	45.0	34.8	28.3	67.3	64.6	60.7	36.7	45.5	57.5	56.4	68.0	67.0	60.0	22.1	47.9	68.0	59.1	60.7
	TDSR	$100k : 1 : 0 + 200k : 1 : 0.01$	24.06	62.2	70.6	70.1	55.0	49.4	29.8	71.4	71.1	74.4	41.3	62.6	66.4	69.8	76.1	71.7	67.7	32.8	59.9	71.8	70.9	62.0
	LR	-	-	16.6	23.8	17.6	12.2	11.3	9.09	24.6	26.1	23.5	6.27	14.3	13.7	20.1	20.5	23.5	20.6	9.53	10.3	16.2	15.0	12.9
	Bicubic	-	21.85	11.2	13.6	9.80	10.9	1.71	9.09	12.3	18.9	22.7	9.09	7.41	9.91	18.8	10.8	16.9	16.1	2.42	9.09	5.67	2.60	16.1
	SRGAN	-	18.72	13.4	27.2	10.1	12.3	9.96	6.13	15.8	15.6	15.6	9.39	9.89	8.16	18.6	11.7	13.0	20.5	9.44	10.8	17.1	6.59	19.9
$8\times$	DBPN	-	17.50	10.6	25.0	9.09	10.8	9.54	0.80	16.3	14.7	13.6	3.45	9.09	7.56	12.2	9.09	9.49	13.52	1.96	9.09	16.1	4.55	16.69
	SR-FT	$100k : 0$	22.77	22.0	32.0	19.3	18.0	10.7	9.60	34.9	34.6	26.4	13.0	14.5	25.1	27.0	22.2	26.9	31.0	9.46	10.9	26.7	18.1	30.3
	SR-FT+	$100k : 1 : 0 + 200k : 1 : 0$	22.82	22.9	32.3	24.1	19.7	11.4	9.74	34.8	34.6	27.7	13.3	14.5	24.5	26.7	23.3	28.8	31.9	9.58	11.3	30.1	18.4	30.8
	TDSR	$100k : 1 : 0 + 200k : 1 : 0.01$	22.26	37.5	49.3	40.9	30.9	25.9	11.4	51.6	47.8	45.0	15.2	31.5	44.1	41.9	50.3	45.6	47.0	14.4	30.6	46.3	40.3	39.6

Table 2. Results on MSCOCO `val2017`. The bracket values is for ($4\times : 8\times$) respectively.

	HR	LR	Bicubic	DBPN	SRGAN	SR-FT	SR-FT+	TDSR
AP@[IoU = 0.50 : 0.95 | area= all]	24.2	(8.2 : 1.9)	(8.1 : 2.0)	(1.2 : 0.1)	(0.6 : 0.1)	(13.7 : 4.4)	(14.1 : 4.8)	(16.7 : 9.8)
AP@[IoU = 0.50 | area= all]	42.2	(15.5 : 4.1)	(14.9 : 3.7)	(2.3 : 0.2)	(1.2 : 0.1)	(24.8 : 8.1)	(25.4 : 8.8)	(30.2 : 18.8)
AP@[IoU = 0.75 | area= all]	24.6	(7.9 : 1.7)	(7.8 : 1.9)	(1.1 : 0.0)	(0.6 : 0.0)	(13.7 : 4.3)	(14.0 : 4.7)	(16.7 : 9.2)
AP@[IoU = 0.50 : 0.95 | area= small]	7.2	(0.2 : 0.0)	(0.9 : 0.1)	(0.1 : 0.0)	(0.1 : 0.0)	(2.0 : 0.3)	(2.2 : 0.3)	(2.7 : 0.7)
AP@[IoU = 0.50 : 0.95 | area= medium]	26.7	(3.8 : 0.4)	(6.5 : 1.2)	(0.9 : 0.0)	(0.4 : 0.1)	(12.8 : 3.3)	(13.2 : 3.6)	(15.8 : 6.7)
AP@[IoU = 0.50 : 0.95 | area= large]	39.4	(19.9 : 4.7)	(17.6 : 5.2)	(2.7 : 0.1)	(1.5 : 0.1)	(27.2 : 11.0)	(28.0 : 11.4)	(31.0 : 20.8)

were used for testing in all experiments. The input of DBPN was a LR image that was obtained by bicubic downscaling the original (HR, 300×300) image from the data set with a particular scaling factor (i.e., $1/4$ or $1/8$ in our experiments, corresponding to $4\times$ and $8\times$ SR).

Training Setting. We used a batch size of 6. The learning rate was initialized to $1e-4$ for all layers and decreased by a factor of 10 after 2×10^5 iterations for training runs consisting of 300,000 iterations. For optimization, we used Adam with momentum set to 0.9. All experiments were conducted using PyTorch 0.3.1 on NVIDIA TITAN X GPUs.

4.2 Performance on VOC and COCO Dataset

Table 1 shows detailed results per class for comparing our TDSR method to other SR approaches trained on VOC0712 `trainval` and evaluated on VOC2007 `test`, including the baseline bicubic SR, and a recently proposed state-of-the-art SR method (SRGAN [10]). Comparison to SRGAN is particularly interesting since it uses a different kind of objective (adversarial/perceptual) which may be assumed to be better suited for task-driven SR. Note that all the other SR models were just pretrained, and not fine-tuned on Pascal. We also compared results obtained directly from LR images (padded with black to fit to the pretrained SSD300 detector). It is shown that SR-FT+ successfully to have highest PSNR. However, TDSR overpowered other methods for all classes and boosted the performance of LR images.

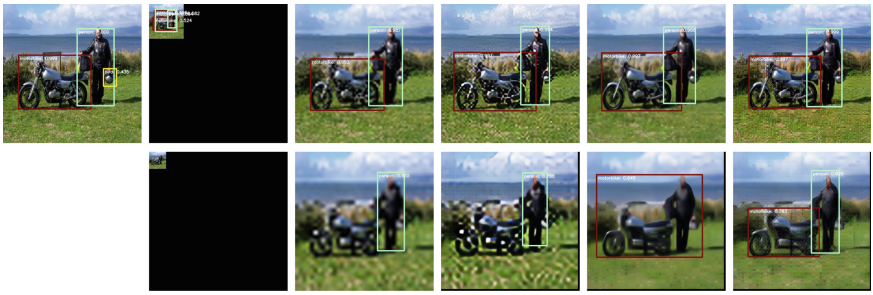
We see that reduction in resolution has a drastic effect on the AP of the detector, dropping it from 75.8 to 41.7 for $4\times$ and 16.6 for $8\times$ as shown in Table 1. This is presumably due to both the actual loss of information, and the limitations of the detector

architecture which may miss small bounding boxes. The performance is not significantly improved by non-task-driven SR methods, which in some cases actually harm it further! However, our proposed TDSR approach obtains significantly better results for both scaling factors, and recovers a significant fraction of the detection accuracy lost in LR.

In accordance with VOC results, the results trained on COCO dataset is also shown the effectiveness of TDSR. Table 2 shows detailed result on COCO `eval2017`. TDSR is successfully to increase the accuracy of LR images roughly by 100% and 500% for $4\times$ and $8\times$, respectively and outperform other methods. TDSR consistently has better performance than SR-FT+ for most of the classes especially on $8\times$.

4.3 Qualitative Analysis

Figures. 3, 4, and 5 show examples of our results compared with those of other methods. The results for SRGAN ([10]) and SR-FT+ sometimes confuse the detector and recognize it as different object classes, again indicating that optimizing L_{rec} and high PSNR do not necessarily correlate with the accuracy. Meanwhile, unique pattern that produced by our proposed optimization helps the detector to recognize the objects better. Note that the TDSR does produce, in many images, artifacts somewhat reminiscent of those in DeepDream ([12]), but those are mild, and are offset by a drastically increased detection accuracy.



(a) HR (b) LR (c) Bicubic (d) SRGAN (e) SR-FT+ (f) TDSR

Fig. 3. Sample results for $4\times$ (upper row) and $8\times$ (lower row).

(a) HR+Blur (b) LR (c) Bicubic (d) SR-FT+ (e) TDSR

Fig. 4. Sample results on blur images for $8\times$ (lower row).

(a) HR+Noise (b) LR (c) Bicubic (d) SR-FT+ (e) TDSR

Fig. 5. Sample results on noise images for $4\times$ (upper row).

5 Conclusions

We have proposed a simple yet effective objective for training SR: a compound loss that caters to the downstream semantic task, and not just to the pixel-wise image reconstruction task as traditionally done. Our results, which consistently exceed alternative SR methods in all conditions, indicate that modern end-to-end training enables joint optimization of tasks what has traditionally been separated into low-level vision (super-resolution) and high-level vision (object detection).

References

1. Agustsson, E., Timofte, R.: Ntire 2017 challenge on single image super-resolution: dataset and study. In The IEEE Conference on Computer Vision and Pattern Recognition (CVPR) Workshops (July 2017)
2. Dosovitskiy, A., Brox, T.: Generating images with perceptual similarity metrics based on deep networks. In: Advances in Neural Information Processing Systems, pp. 658–666 (2016)
3. Goodfellow, I., et al.: Generative adversarial nets. In: Advances in Neural Information Processing Systems, pp. 2672–2680 (2014)
4. Hanhart, P., Korshunov, P., Ebrahimi, T.: Benchmarking of quality metrics on ultra-high definition video sequences. In: 2013 18th International Conference on Digital Signal Processing (DSP), pp. 1–8. IEEE (2013)
5. Haris, M., Shakhnarovich, G., Ukita, N. Deep back-projection networks for super-resolution. In: IEEE Conference on Computer Vision and Pattern Recognition (2018)
6. Hu, P., Ramanan, D.: Finding tiny faces. In: 2017 IEEE Conference on Computer Vision and Pattern Recognition (CVPR), pp. 1522–1530. IEEE (2017)
7. Johnson, J., Alahi, A., Fei-Fei, L.: Perceptual losses for real-time style transfer and super-resolution. In: Leibe, B., Matas, J., Sebe, N., Welling, M. (eds.) ECCV 2016. LNCS, vol. 9906, pp. 694–711. Springer, Cham (2016). https://doi.org/10.1007/978-3-319-46475-6_43
8. Kong, S., Fowlkes, C.: Recurrent scene parsing with perspective understanding in the loop. In: Proceedings of 2018 IEEE Conference on Computer Vision and Pattern Recognition (CVPR) (2018)
9. Kundu, D., Evans, B.L.: Full-reference visual quality assessment for synthetic images: a subjective study. In: 2015 IEEE International Conference on Image Processing (ICIP), pp. 2374–2378. IEEE (2015)
10. Ledig, C., et al. Photo-realistic single image super-resolution using a generative adversarial network. In: IEEE Conference on Computer Vision and Pattern Recognition (CVPR) (July 2017)

11. Liu, W., et al.: SSD: single shot multibox detector. In: Leibe, B., Matas, J., Sebe, N., Welling, M. (eds.) ECCV 2016. LNCS, vol. 9905, pp. 21–37. Springer, Cham (2016). https://doi.org/10.1007/978-3-319-46448-0_2
12. Mordvintsev, A., Tyka, M., Olah, C.: Inceptionism: Going deeper into neural networks. https://research.googleblog.com/2015/06/inceptionism-going-deeper-into-neural.html (June 2015)
13. Nasrollahi, K., Moeslund, T.B.: Super-resolution: a comprehensive survey. Mach. Vis. Appl. **25**(6), 1423–1468 (2014). https://doi.org/10.1007/s00138-014-0623-4
14. Sajjadi, M.S., Schölkopf, B., Hirsch, M.: Enhancenet: single image super-resolution through automated texture synthesis. In: 2017 IEEE International Conference on Computer Vision (ICCV), pp. 4501–4510. IEEE (2017)
15. Wang, Z., Bovik, A.C., Sheikh, H.R., Simoncelli, E.P.: Image quality assessment: from error visibility to structural similarity. IEEE Trans. Image Process. **13**(4), 600–612 (2004)
16. Yang, C.-Y., Ma, C., Yang, M.-H.: Single-image super-resolution: a benchmark. In: Fleet, D., Pajdla, T., Schiele, B., Tuytelaars, T. (eds.) ECCV 2014. LNCS, vol. 8692, pp. 372–386. Springer, Cham (2014). https://doi.org/10.1007/978-3-319-10593-2_25
17. Yu, X., Porikli, F.: Ultra-resolving face images by discriminative generative networks. In: Leibe, B., Matas, J., Sebe, N., Welling, M. (eds.) ECCV 2016. LNCS, vol. 9909, pp. 318–333. Springer, Cham (2016). https://doi.org/10.1007/978-3-319-46454-1_20