On-Demand Learning for Deep Image Restoration

THE UNIVERSITY OF TEXAS AT AUSTIN

Ruohan Gao and Kristen Grauman, University of Texas at Austin
http://vision.cs.utexas.edu/projects/on_demand_learning/

ICCV17
International Conference on Computer Vision 2017

BACKGROUND

Inpainting

size of inpainting area

Deblurring

degree of blurriness

Pixel Interpolation

percentage of deleted pixels

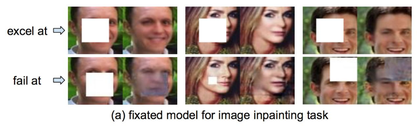
Denoising

severity of noise

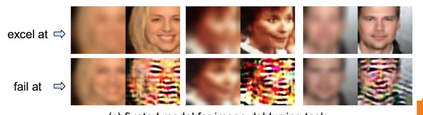
Image corruption exists in various degrees of severity!

THE FIXATION PROBLEM

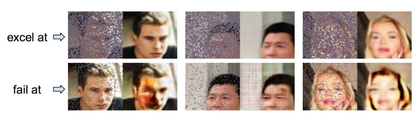
Status quo is to train a learning algorithm to restore images with a controlled degree of corruption. E.g., Yeh *et al.* 2017, Liu e*t al.* 2016, Burger *et al.* 2012...



(a) fixated model for image inpainting task



(c) fixated model for image deblurring task



(b) fixated model for pixel interpolation task

(d) fixated model for image denoising task

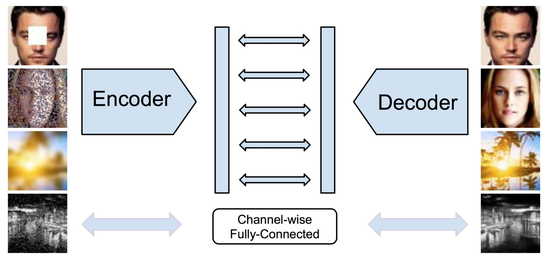
Existing methods risk training "fixated models" --- models that perform well only at a particular level of difficulty.

PROBLEM FORMULATION

Objective: *Train an image restoration system to succeed across a spectrum of difficulty levels.*

- real images: $\{\mathcal{R}_i\}$
- corrupted images: $\{\mathcal{C}_i\}$
- a general encoder-decoder deep network f with weights $\mathbf{w}$
- objective:

$$\hat{\mathbf{w}} = \underset{\mathbf{w}}{\operatorname{argmin}} \sum_i ||\mathcal{R}_i - f(\mathcal{C}_i, \mathbf{w})||_2^2$$



ON-DEMAND LEARNING ALGORITHM

Key Idea: *Let the system guide its own learning towards the right proportion of sub-tasks per difficulty level by creating a feedback mechanism to self-generate instances where they are needed most.*

On-demand learning:

N sub-tasks of increasing difficulty: $T_1, T_2, \ldots, T_N$
# of training examples for sub-task T_i per batch: B_i
Batch Size: $\mathbb{B}$

Algorithm 1: On-Demand Learning

Initialization: $B_i = \mathbb{B}/N$
while *time budget has not run out* **do**
  continue training for one epoch and snapshot;
  if *end of epoch* **then**
    $i = 1$;
    for $i \leq N$ **do**
      validate snapshot model on sub-task T_i;
      get mean PSNR P_i;
    end
    update $B_i = \frac{1/P_i}{\sum_{i=1}^{N} 1/P_i} \cdot \mathbb{B}$;
  end
end

Other training paradigms:

- ➔ Rigid joint: simply pool training instances across difficulty levels
- ➔ Curriculum learning: order training samples from easy to hard

Algorithm 2: Staged Curriculum Learning

Initialization: $B_1 = \mathbb{B}, B_{2,\ldots,N} = 0, i = 1$
while $i <= N$ **do**
  train using $\frac{1}{N}$ time budget;
  $i = i + 1$;
  $B_i = \mathbb{B}, B_{\{1,\ldots,N\}-\{i\}} = 0$;
end

Algorithm 3: Cumulative Curriculum Learning

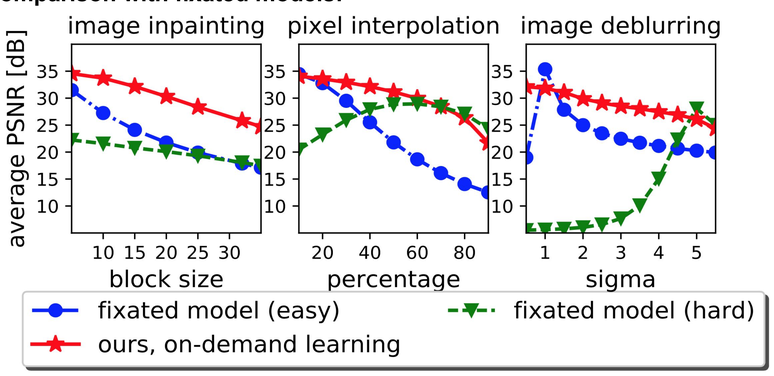
Initialization: $B_1 = \mathbb{B}, B_{2,\ldots,N} = 0, i = 1$
while $i <= N$ **do**
  train using $\frac{1}{N}$ time budget;
  $i = i + 1$;
  update $B_{\{1,\ldots,i\}} = \mathbb{B}/i$;
end

EXPERIMENT RESULTS

Experiment set-up: five difficulty levels (N=5)

- Image Inpainting: inpainting size 1x1-6x6, 7x7-12x12, 13x13-18x18, 19x19-24x24, 25x25-30x30
- Pixel Interpolation: percentage of deleted pixels 0%-15%, 15%-30%, 30%-45%, 45%-60%, 60%-75%
- Image Deblurring: blur kernel width 0-1, 1-2, 2-3, 3-4, 4-5
- Image Denoising: variance of AWG noise 0-20, 20-40, 40-60, 60-80, 80-100

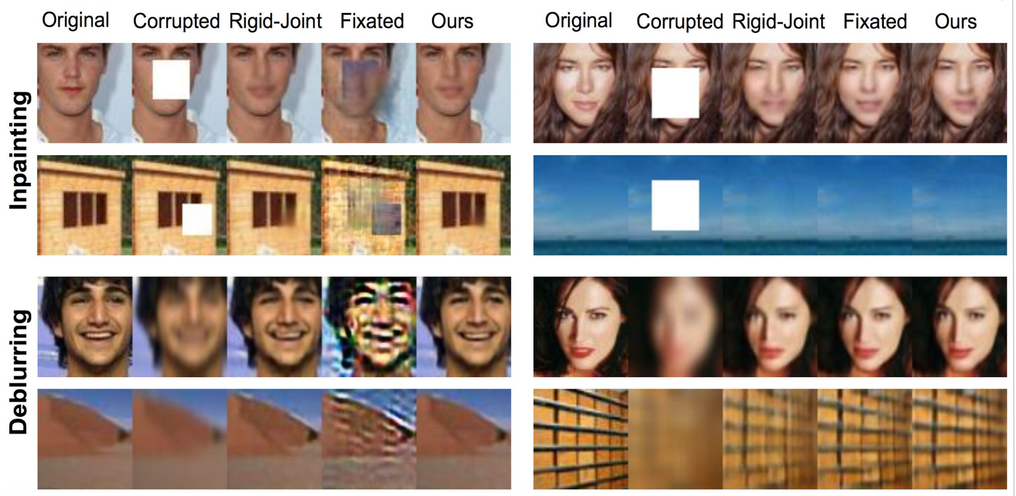
Comparison with fixated models:



Observation: *Our algorithm performs well over the spectrum of difficulty, whereas fixated models perform well at only a certain level of corruption.*

Comparison of overall performance:

PSNR (in dB)	CelebA Dataset				SUN397 Dataset			
	Deblurring	Pixel Interpolation	Inpainting	Denoising	Deblurring	Pixel Interpolation	Inpainting	Denoising
Rigid-Joint Learning	29.40	31.86	32.11	26.38	28.53	31.98	31.13	25.69
Cumulative Curriculum	28.70	31.68	31.47	26.31	27.86	31.70	30.75	25.60
Staged Curriculum	15.59	28.51	31.30	24.74	14.44	28.13	30.42	23.87
Hard Mining	27.33	29.15	29.47	25.10	26.35	29.01	29.83	24.10
On-Demand Learning	**29.58**	**32.09**	**32.30**	**26.48**	**28.70**	**32.21**	**31.38**	**25.80**

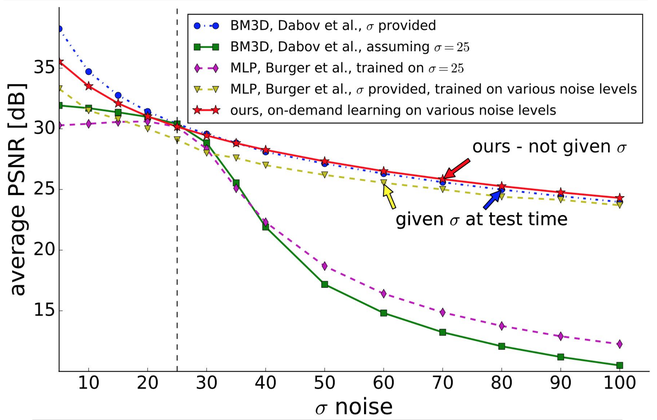


Observation: *By automatically guiding the balance among sub-tasks, on-demand learning successfully addresses the fixation problem, and obtains the best all-around performance.*

Comparison to existing inpainting and denoising methods:

Method	Central Square Block		Arbitrary Square Block	
	L2 Loss	PSNR	L2 Loss	PSNR
Pathak *et al.* (CVPR 2016) Center	**0.83%**	**22.16 dB**	6.84%	11.80 dB
Pathak *et al.* (CVPR 2016) + Rand drop	2.47%	16.18 dB	2.51%	16.20 dB
Ours	0.93%	20.74 dB	**1.04%**	**20.31 dB**

Observation: *The inpainter trained under our framework not only can inpaint central square blocks, but does similarly well when tested on square blocks located anywhere in the image.*



Observation: *Our image denoising system consistently performs well on all noise levels, yet we do not assume knowledge of noise level during testing.*