Advanced Deep Learning Architectures
COMP 5214 & ELEC 5680

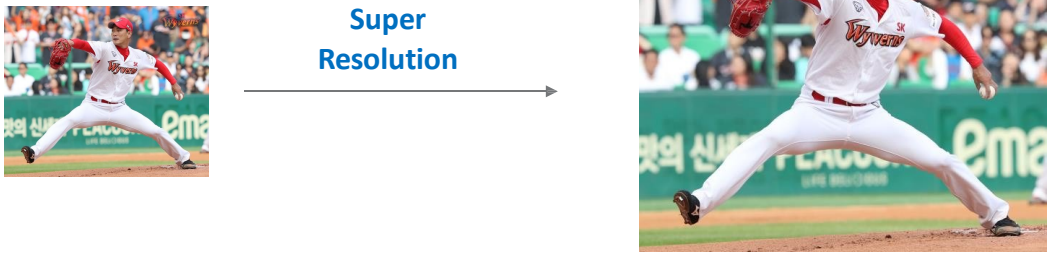
Instructor: Dr. Qifeng Chen
https://cqf.io

Single Image Super Resolution

Credit: Ho Seong Lee

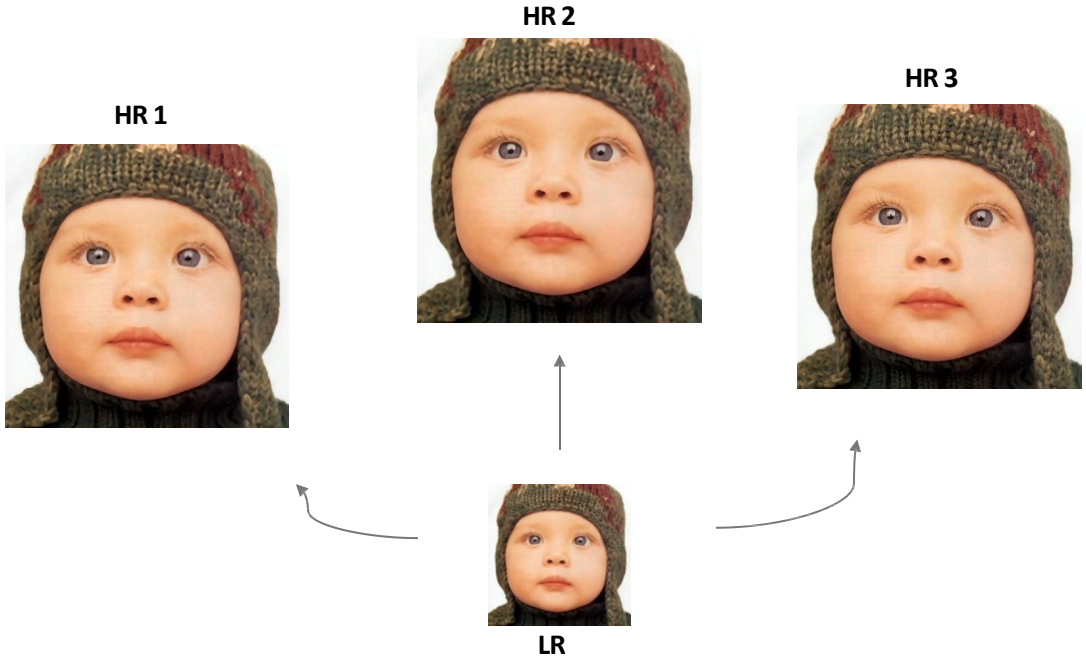
- What is Super Resolution?
- Applications of Super Resolution
- Deep Learning for Single Image Super Resolution
- Some Issues for Super Resolution

- Super Resolution
  - Restore High-Resolution(HR) image(or video) from Low-Resolution(LR) image(or video)
  - According to the number of input LR images, SR can be classified **SISR** or MISR
    - Single Image Super Resolution



What is Super Resolution?

- Single Image Super Resolution
  - Restore High-Resolution(HR) image(or video) from Low-Resolution(LR) image (or video)
  - Ill-Posed Problem.. (Regular Inverse Problem) → We can't have ground truth from LR image



What is Super Resolution?

- Interpolation-based Single Image Super Resolution
  - In image upscaling task, **bicubic** or **bilinear** or **Lanczos** interpolation is usually used.
  - Fast, easy.. but low quality..

Super Resolution

Deep SR

bilinear

What is Super Resolution?

- Single Image Super Resolution algorithms
  - Interpolation-based method
  - Reconstruction-based method
  - **(Deep) Learning-based method**

- Today, I will cover **learning-based method**

Applications of Super Resolution

- Satellite image processing
- Medical image processing
- Multimedia Industry and Video Enhancement

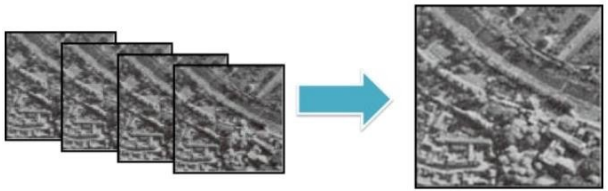


Fig 1: SR for satellite image [22]

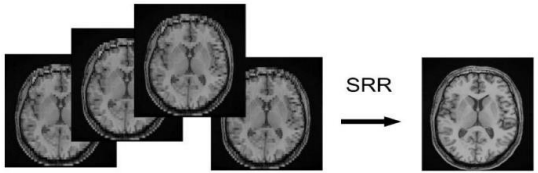


Fig 2: SR in Medical Imaging [23]

TV & Monitor

Reference: "Super Resolution Applications in Modern Digital Image Processing", 2016 IJCA

- Learning-based Single Image Super Resolution
  - For tackling regular inverse problem, almost use this paradigm
  - **[HR(GT) image]** + **[distortion & down-sampling]** → [**LR(input) image]**
    - This is limitation of SISR training
    - Overall restoration quality is dependent on the **distortion & down-sampling** method

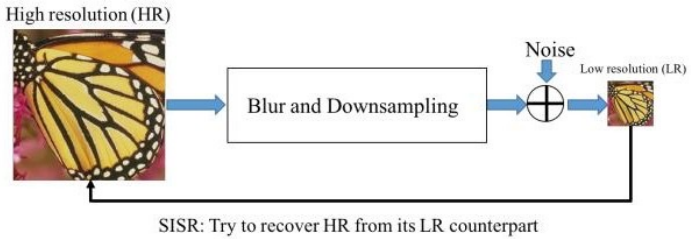


Figure 1: Sketch of the overall framework of SISR.

Reference: "Deep Learning for Single Image Super-Resolution: A Brief Review", 2018 IEEE Transactions on Multimedia (TMM)

- Learning-based Single Image Super Resolution
  - **[HR(GT) image]** + **[distortion & down-sampling]** → **[LR(input) image]**
    - In CVPR 2017 SR Challenge, many team showed many degradation of quality metric

		Track 1: bicubic downscaling						Track 2: unknown downscaling					
		×2		×3		×4		×2		×3		×4	
Team	User	PSNR	SSIM	PSNR	SSIM	PSNR	SSIM	PSNR	SSIM	PSNR	SSIM	PSNR	SSIM
SNU_CVLab [1]	limbee	34.93$_{(1)}$	0.948	31.13$_{(1)}$	0.889	26.91*$_{(14)}$	0.752*	34.00$_{(1)}$	0.934	30.78$_{(1)}$	0.881	28.77$_{(1)}$	0.826
SNU_CVLab [2]	sanghyun	34.83$_{(2)}$	0.947	31.04$_{(2)}$	0.888	29.04$_{(1)}$	0.836	33.86$_{(2)}$	0.932	30.67$_{(2)}$	0.879	28.62$_{(2)}$	0.821
HelloSR	sparkfirer	34.47$_{(4)}$	0.944	30.77$_{(4)}$	0.882	28.82$_{(3)}$	0.830	33.67$_{(3)}$	0.930	30.51$_{(3)}$	0.876	28.54$_{(3)}$	0.819
Lab402	iorism	34.66$_{(3)}$	0.946	30.83$_{(3)}$	0.884	28.83$_{(2)}$	0.830	32.92$_{(7)}$	0.921	30.31$_{(4)}$	0.871	28.14$_{(6)}$	0.807
VICLab	JSChoi	34.29$_{(5)}$	0.943	30.52$_{(5)}$	0.880	28.55$_{(5)}$	0.845						
UIUC-IFP	fyc0624	34.19$_{(6)}$	0.942	30.44$_{(7)}$	0.877	28.49$_{(6)}$	0.821	28.54$_{(14)}$	0.840	28.11$_{(14)}$	0.816	24.96$_{(15)}$	0.717
HIT-ULSee	chenyunjin	34.07$_{(7)}$	0.941	30.21$_{(9)}$	0.871	28.49$_{(6)}$	0.822	33.40$_{(4)}$	0.927	30.21$_{(6)}$	0.871	28.30$_{(4)}$	0.812
I hate mosaic	tzm1003306213	34.05$_{(8)}$	0.940	30.47$_{(6)}$	0.878	28.59$_{(4)}$	0.824						
nicheng	nicheng									30.24$_{(5)}$	0.871	28.26$_{(5)}$	0.811
GTY	giangbui	34.03$_{(9)}$	0.941	30.24$_{(8)}$	0.874	28.34$_{(7)}$	0.817	33.32$_{(5)}$	0.926	30.14$_{(7)}$	0.869	27.33$_{(8)}$	0.785
DL-61-86	rosinwang							33.10$_{(6)}$	0.922	30.05$_{(8)}$	0.863	28.07$_{(7)}$	0.800
faceall_Xlabs	xjc_faceall	33.73$_{(10)}$	0.937	30.07$_{(10)}$	0.869	27.99$_{(10)}$	0.805	24.98$_{(15)}$	0.707	29.87$_{(9)}$	0.862	26.84$_{(10)}$	0.762
SR2017	xiangyu_xu	33.54$_{(11)}$	0.934	29.89$_{(12)}$	0.865	28.07$_{(8)}$	0.809	29.92$_{(12)}$	0.871	28.84$_{(11)}$	0.836	26.05$_{(11)}$	0.754
SDQ_SR	XibinSong	33.49$_{(12)}$	0.936					32.35$_{(8)}$	0.912				
HCILab	phunghx	33.47$_{(13)}$	0.934	29.92$_{(11)}$	0.866	28.03$_{(9)}$	0.807	31.13$_{(9)}$	0.896	29.26$_{(10)}$	0.849	25.96$_{(12)}$	0.749
iPAL	antonGo	33.42$_{(14)}$	0.932	29.89$_{(12)}$	0.865	27.99$_{(10)}$	0.806						
WSDSR	cristovao.a.cruz	33.19$_{(15)}$	0.933	29.74$_{(13)}$	0.864	27.92$_{(11)}$	0.805						
Resonance	arnavkj95							30.21$_{(10)}$	0.889	28.43$_{(13)}$	0.840	24.79$_{(16)}$	0.724
zrfanzy	zrfan	31.87$_{(17)}$	0.927	28.80$_{(15)}$	0.858	27.67$_{(12)}$	0.800	21.94$_{(16)}$	0.618	18.03$_{(15)}$	0.490	26.95$_{(9)}$	0.773
assafsho	assafsho	30.39$_{(18)}$	0.894	27.23$_{(16)}$	0.806	25.74$_{(15)}$	0.742						
UESTC-kb545	naiven											25.08$_{(14)}$	0.714
spectrum	spectrum							28.76$_{(13)}$	0.854				
bicubic interp.	*baseline*	31.01	0.900	28.22	0.822	26.65	0.761	25.08	0.713	25.81	0.736	21.84	0.583

Table 2. NTIRE 2017 Challenge results and final rankings on DIV2K test data. (*) the checked SNU_CVLab[1] model achieved 29.09dB PSNR and 0.837 SSIM.

Reference: http://www.vision.ee.ethz.ch/~timofter/publications/NTIRE2017SRchallenge_factsheets.pdf

Deep Learning for Single Image Super Resolution

- First Deep Learning architecture for Single Image Super Resolution
  - SRCNN(2014) – three-layer CNN, MSE Loss, **Early upsampling**
  - Compared to traditional methods, it shows excellent performance.

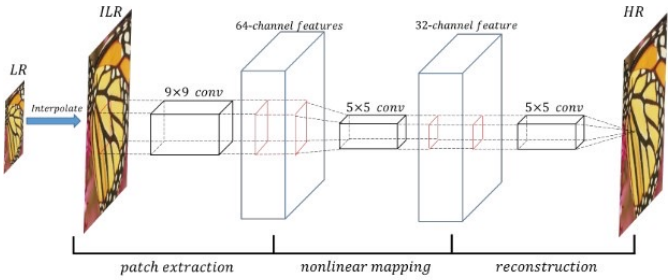
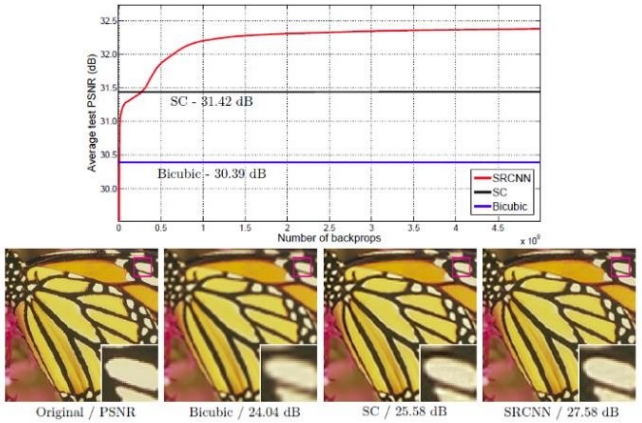


Figure 2: Sketch of the SRCNN architecture.



Reference: "Image Super-Resolution Using Deep Convolutional Networks", 2014 ECCV

- Efficient Single Image Super Resolution
  - FSRCNN(2016), ESPCN(2016)
  - Use **Late Upsampling** with deconvolution or sub-pixel convolutional layer

Inefficient in Memory, FLOPS

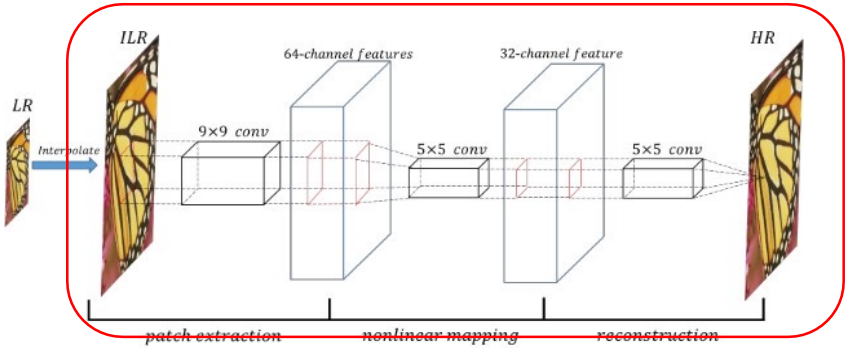
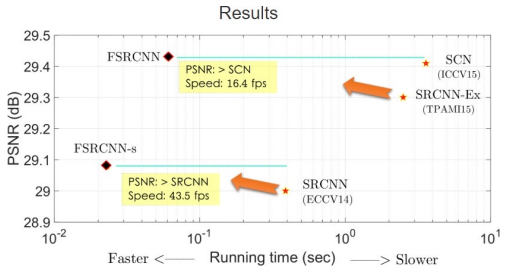


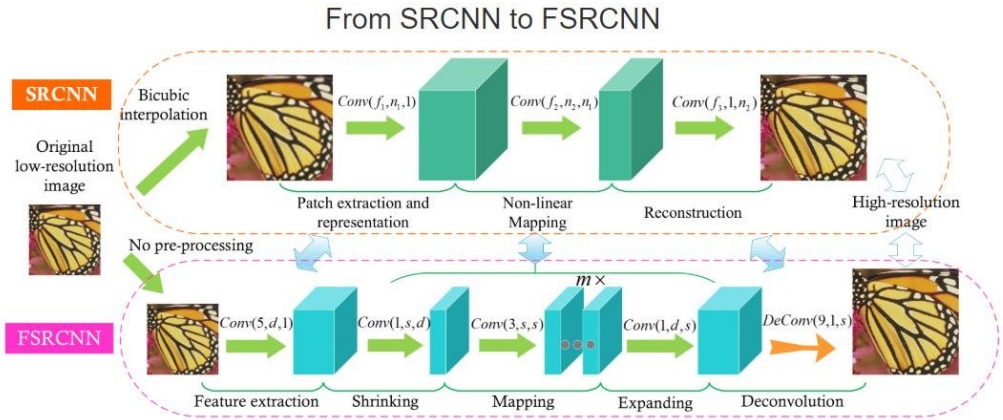
Figure 2: Sketch of the SRCNN architecture.

Reference: "Image Super-Resolution Using Deep Convolutional Networks", 2014 ECCV

Deep Learning for Single Image Super Resolution

- FSRCNN(Fast Super-Resolution Convolutional Neural Network)
  - Use Deconvolution layer instead of pre-processing(upsampling)
  - Faster and more accurate than SRCNN





Reference: "Accelerating the Super-Resolution Convolutional Neural Network", 2016 ECCV

Deep Learning for Single Image Super Resolution

- ESPCN(Real-Time Single Image and Video Super-Resolution Using an Efficient Sub-Pixel Convolutional Neural Network)
  - Use sub-pixel convolutional layer (pixel shuffler or depth_to_space)
  - This sub-pixel convolutional layer is used in recent SR models

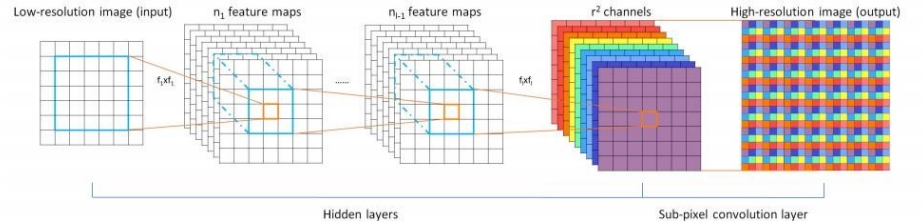


Figure 1. The proposed efficient sub-pixel convolutional neural network (ESPCN), with two convolution layers for feature maps extraction, and a sub-pixel convolution layer that aggregates the feature maps from LR space and builds the SR image in a single step.

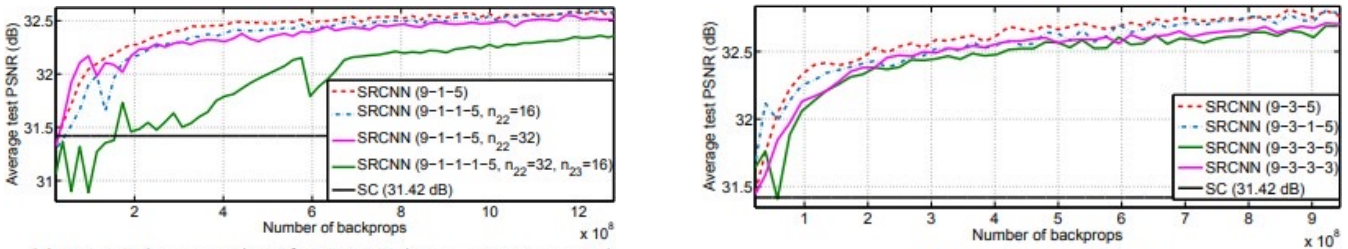
```
class Net(nn.Module):
    def __init__(self, upscale_factor):
        super(Net, self).__init__()

        self.conv1 = nn.Conv2d(1, 64, (5, 5), (1, 1), (2, 2))
        self.conv2 = nn.Conv2d(64, 64, (3, 3), (1, 1), (1, 1))
        self.conv3 = nn.Conv2d(64, 32, (3, 3), (1, 1), (1, 1))
        self.conv4 = nn.Conv2d(32, 1 * (upscale_factor ** 2), (3, 3), (1, 1), (1, 1))
        self.pixel_shuffle = nn.PixelShuffle(upscale_factor)

    def forward(self, x):
        x = F.tanh(self.conv1(x))
        x = F.tanh(self.conv2(x))
        x = F.tanh(self.conv3(x))
        x = F.sigmoid(self.pixel_shuffle(self.conv4(x)))
        return x
```

Reference: “Real-Time Single Image and Video Super-Resolution Using an Efficient Sub-Pixel Convolutional Neural Network”, 2016 CVPR
Code: https://github.com/leftthomas/ESPCN

- Deeper Networks for Super-Resolution
  - SRCNN, FSRCNN, ESPCN are shallow network → Why not deep network?
  - Failed to train deeper models.. → Use shallow network → how to use deeper network?



(a) 9-1-1-5 ($n_{22} = 32$) and 9-1-1-1-5 ($n_{22} = 32, n_{23} = 16$)

(b) 9-3-3-5 and 9-3-3-3

Fig. 9. Deeper structure does not always lead to better results.

Reference: "Image Super-Resolution Using Deep Convolutional Networks", 2014 ECCV

Deep Learning for Single Image Super Resolution

- VDSR(Accurate Image Super-Resolution Using Very Deep Convolutional Networks)
  - VGG based deeper model(20-layer) for Super-Resolution → large receptive field
  - Residual learning & High learning rate with gradient clipping
  - MSE Loss, **Early upsampling**

Epoch	10	20	40	80
Residual	36.90	36.64	37.12	37.05
Non-Residual	27.42	19.59	31.38	35.66
Difference	9.48	17.05	5.74	1.39

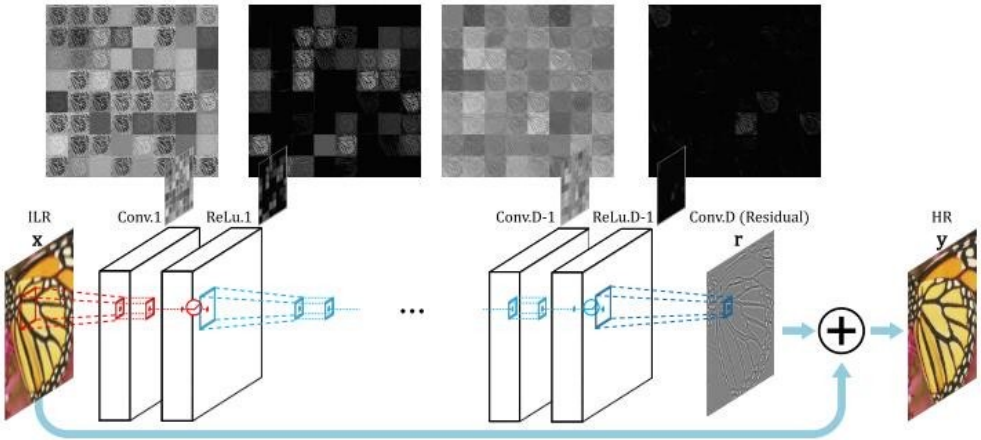
(a) Initial learning rate 0.1

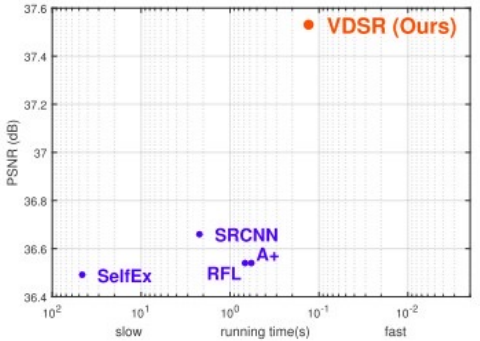
Epoch	10	20	40	80
Residual	36.74	36.87	36.91	36.93
Non-Residual	30.33	33.59	36.26	36.42
Difference	6.41	3.28	0.65	0.52

(b) Initial learning rate 0.01

Epoch	10	20	40	80
Residual	36.31	36.46	36.52	36.52
Non-Residual	33.97	35.08	36.11	36.11
Difference	2.35	1.38	0.42	0.40

(c) Initial learning rate 0.001

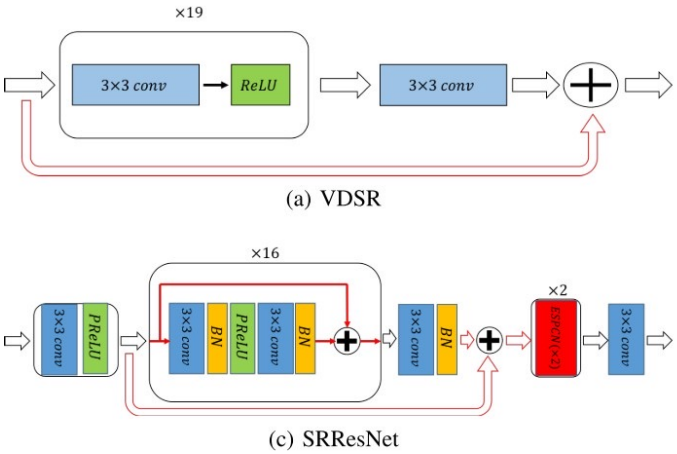




Reference: "Accurate Image Super-Resolution Using Very Deep Convolutional Networks", 2016 CVPR

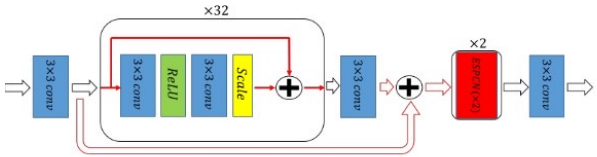
Deep Learning for Single Image Super Resolution

- Deeper Networks for Super-Resolution after VDSR
  - DRCN(Deeply-recursive Convolutional network), 2016 CVPR
  - SRResNet, 2017 CVPR
  - DRRN(Deep Recursive Residual Network), 2017 CVPR

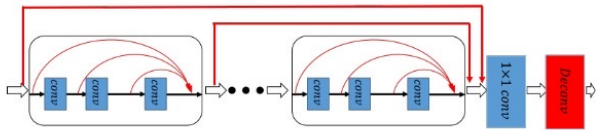


Reference: "Deep Learning for Single Image Super-Resolution: A Brief Review", 2018 IEEE Transactions on Multimedia (TMM)

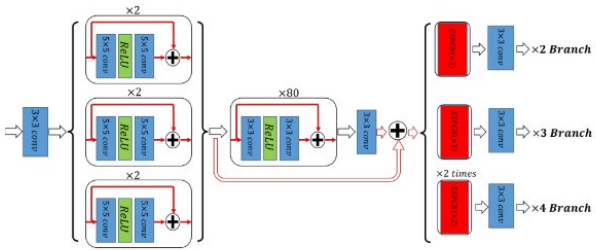
- Deeper Networks for Super-Resolution after VDSR
  - EDSR, MDSR (Enhanced Deep Residual Network, Multi Scale EDSR), 2017 CVPRW
  - DenseSR, 2017 CVPR
  - MemNet, 2017 CVPR



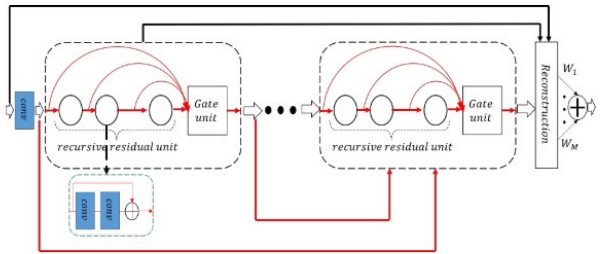
(e) EDSR



(f) DenseSR



(g) MDSR

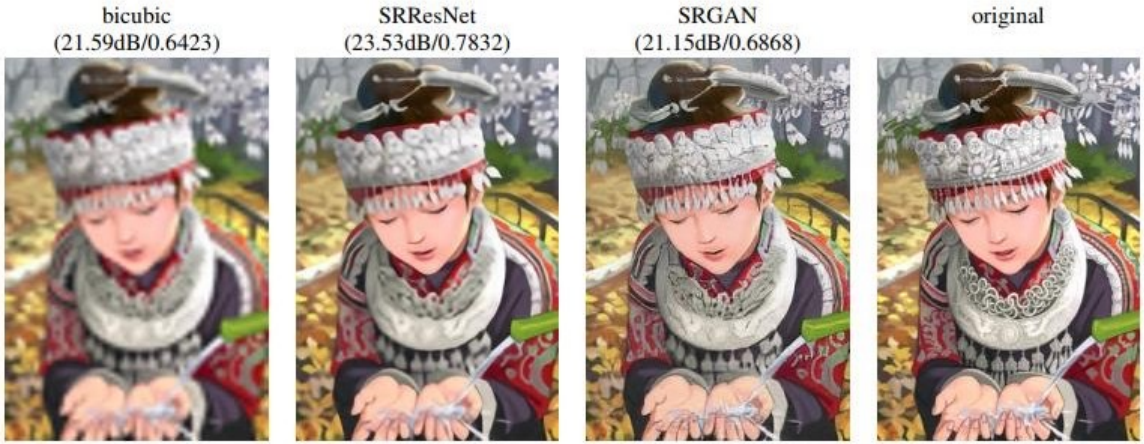


(h) MemNet

Reference: "Deep Learning for Single Image Super-Resolution: A Brief Review", 2018 IEEE Transactions on Multimedia (TMM)

Deep Learning for Single Image Super Resolution

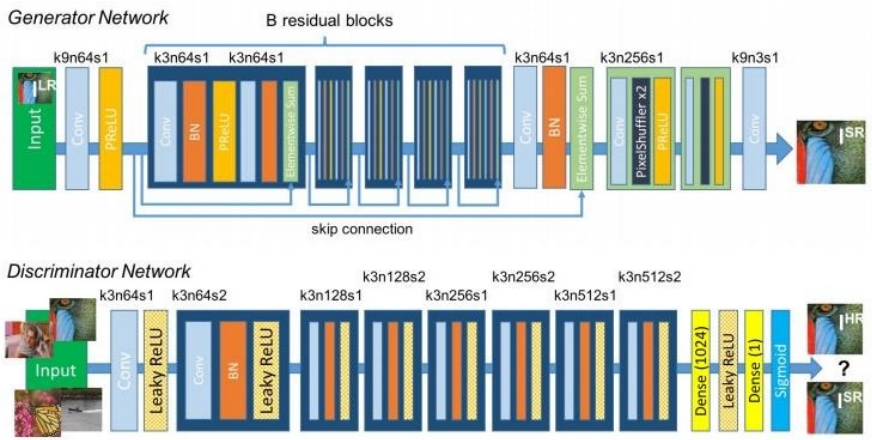
- Generative Adversarial Network(GAN) for Super-Resolution
  - SRGAN(Photo-Realistic Single Image Super-Resolution Using a Generative Adversarial Network)
  - **First GAN-based** SR Model, MSE Loss → Blurry Output → GAN loss + Content loss = **Perceptual loss**



Reference: "Photo-Realistic Single Image Super-Resolution Using a Generative Adversarial Network", 2017 CVPR

Deep Learning for Single Image Super Resolution

- Photo-Realistic Single Image Super-Resolution Using a Generative Adversarial Network
  - MSE Loss → Blurry Output → GAN loss + Content loss = **Perceptual loss**
  - Replace MSE loss to **VGG loss** (used in style transfer) and add **adversarial loss**



$$l^{SR}_{MSE} = \frac{1}{r^2WH}\sum_{x=1}^{rW}\sum_{y=1}^{rH}(I^{HR}_{x,y} - G_{\theta_G}(I^{LR})_{x,y})^2 \quad (4)$$

$$l^{SR}_{VGG/i.j} = \frac{1}{W_{i,j}H_{i,j}}\sum_{x=1}^{W_{i,j}}\sum_{y=1}^{H_{i,j}}(\phi_{i,j}(I^{HR})_{x,y} - \phi_{i,j}(G_{\theta_G}(I^{LR}))_{x,y})^2 \quad (5)$$

$$l^{SR}_{Gen} = \sum_{n=1}^{N} -\log D_{\theta_D}(G_{\theta_G}(I^{LR})) \quad (6)$$

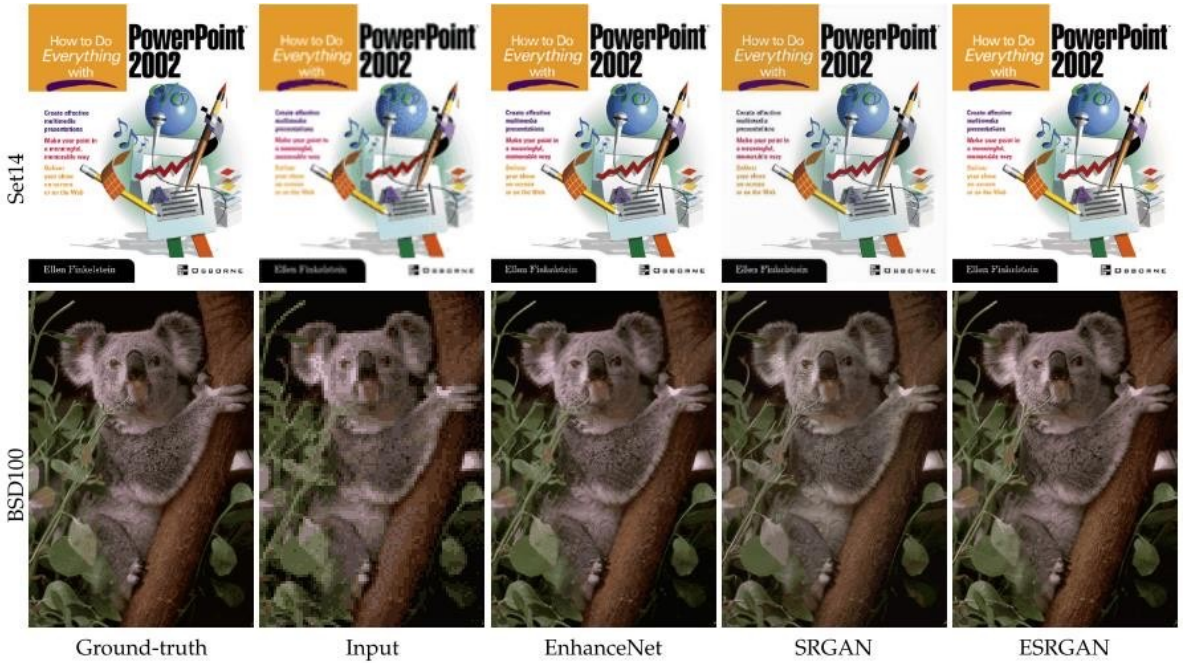
$$l^{SR} = \underbrace{l^{SR}_{X}}_{\text{content loss}} + \underbrace{10^{-3}l^{SR}_{Gen}}_{\text{adversarial loss}} \quad (3)$$

perceptual loss (for VGG based content losses)

Reference: "Photo-Realistic Single Image Super-Resolution Using a Generative Adversarial Network", 2017 CVPR

Deep Learning for Single Image Super Resolution

- Generative Adversarial Network(GAN) for Super-Resolution
  - SRGAN, EnhanceNet, SRFeat, ESRGAN



Reference: "A Deep Journey into Super-resolution: A survey", 2019 arXiv

- Deep Learning for Single Image Super Resolution

TABLE 2
Mean PSNR and SSIM for the SR methods evaluated on the benchmark datasets. The '-' indicates that the method is not suitable to handle the images of the corresponding dataset.

		Set5		Set14		BSD100		Urban100		DIV2K		Manga109	
Scale	Method	PSNR	SSIM	PSNR	SSIM	PSNR	SSIM	PSNR	SSIM	PSNR	SSIM	PSNR	SSIM
×2	Bicubic	33.68	0.9304	30.24	0.8691	29.56	0.8435	26.88	0.8405	32.45	0.904	31.05	0.935
	SRCNN	36.66	0.9542	32.45	0.9067	31.36	0.8879	29.51	0.8946	34.59	0.932	35.72	0.968
	FSRCNN	36.98	0.9556	32.62	0.9087	31.50	0.8904	29.85	0.9009	34.74	0.934	36.62	0.971
	SCN	36.52	0.953	32.42	0.904	31.24	0.884	29.50	0.896	34.98	0.937	35.51	0.967
	REDNet	37.66	0.9599	32.94	0.9144	31.99	0.8974	-	-	-	-	-	-
	VDSR	37.53	0.9587	33.05	0.9127	31.90	0.8960	30.77	0.9141	35.43	0.941	37.16	0.974
	DRCN	37.63	0.9588	33.06	0.9121	31.85	0.8942	30.76	0.9133	35.45	0.940	37.57	0.973
	LapSRN	37.52	0.9591	32.99	0.9124	31.80	0.8949	30.41	0.9101	35.31	0.940	37.53	0.974
	DRRN	37.74	0.9591	33.23	0.9136	32.05	0.8973	31.23	0.9188	35.63	0.941	37.92	0.976
	DnCNN	37.58	0.9590	33.03	0.9128	31.90	0.8961	30.74	0.9139	-	-	-	-
	EDSR	38.11	0.9602	33.92	0.9195	32.32	0.9013	32.93	0.9351	35.03	0.9695	39.10	0.9773
	MDSR	38.11	0.9602	33.85	0.9198	32.29	0.9007	32.84	0.9347	34.96	0.9692	38.96	0.978
	ZSSR	37.37	0.9570	33.00	0.9108	31.65	0.8920	-	-	-	-	-	-
	MemNet	37.78	0.9597	33.28	0.9142	32.08	0.8978	31.31	0.9195	-	-	37.72	0.9740
	CMSC	37.89	0.9605	33.41	0.9153	32.15	0.8992	31.47	0.9220	-	-		
	IDN	37.83	0.9600	33.30	0.9148	32.08	0.8985	31.27	0.9196	-	-	38.02	0.9749
	CNF	37.66	0.9590	33.38	0.9136	31.91	0.8962	-	-	-	-	-	-
	BTSRN	37.75	-	33.20	-	32.05	-	31.63	-	-	-	-	-
	SRMDNF	37.79	0.9601	33.32	0.9159	32.05	0.8985	31.33	0.9204	35.54	0.9414	38.07	0.9761
	D-DBPN	38.09	0.9600	33.85	0.9190	32.27	0.9000	32.55	0.9324	-	-	38.89	0.9775
	SelNet	37.89	0.9598	33.61	0.9160	32.08	0.8984	-	-	-	-	-	-
	CARN	37.76	0.9590	33.52	0.9166	32.09	0.8978	31.92	0.9256	36.04	0.9451	38.36	0.9764
	SRRAM	37.82	0.9592	33.48	0.9171	32.12	0.8983	32.05	0.9264	-	-	-	
	RDN	38.24	0.9614	34.01	0.9212	32.34	0.9017	32.89	0.9353	-	-	39.18	0.9780
	RCAN	38.27	0.9614	34.12	0.9216	32.41	0.9027	33.34	0.9384	36.63	0.9491	39.44	0.9786

Reference: "A Deep Journey into Super-resolution: A survey", 2019 arXiv

Deep Learning for Single Image Super Resolution

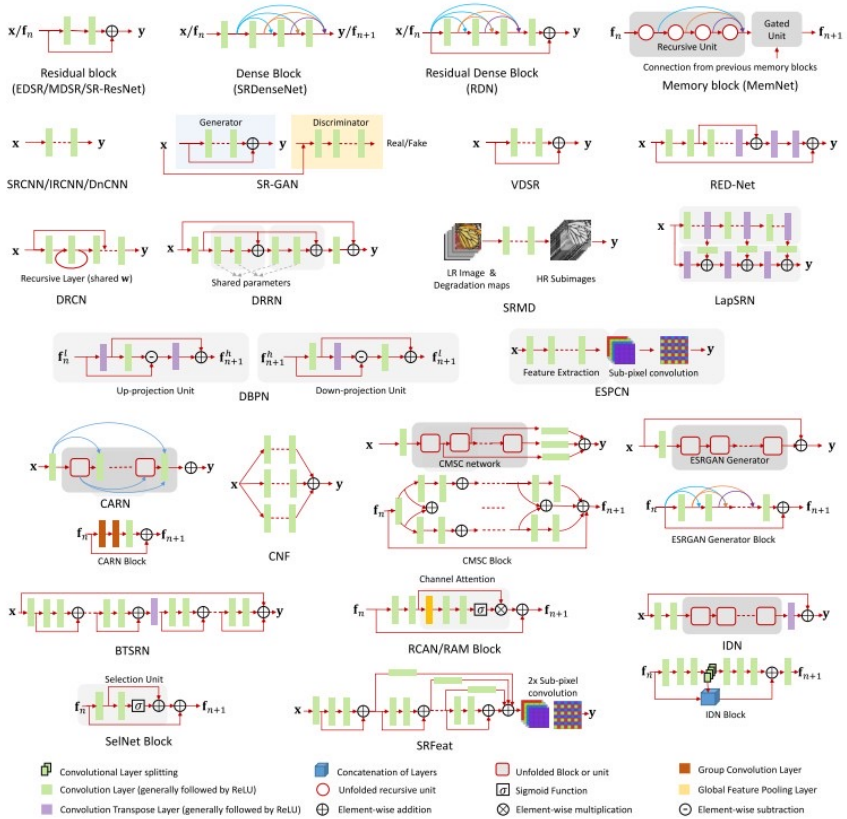
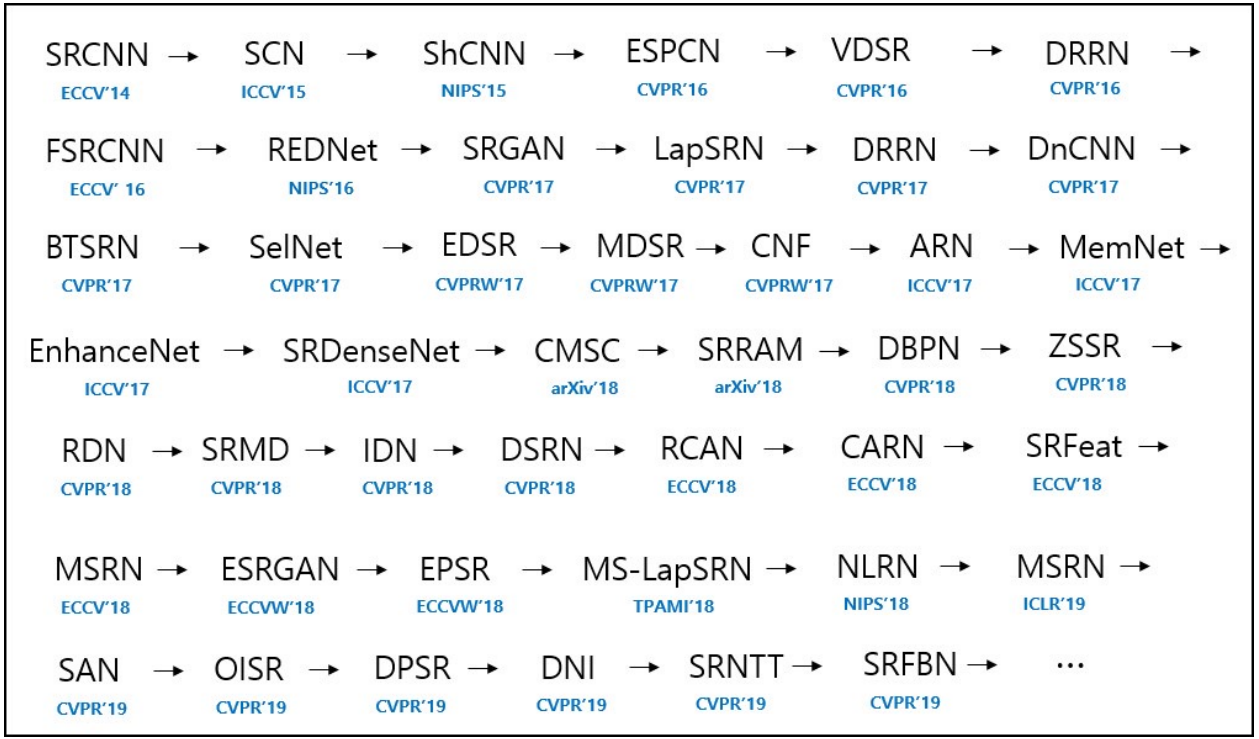


Fig. 2. A glimpse of diverse range of network architectures used for single-image super-resolution using deep networks.

Reference: "A Deep Journey into Super-resolution: A survey", 2019 arXiv

- Deep Learning for Single Image Super Resolution

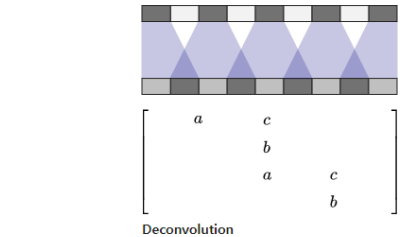


Some Issues for Super Resolution

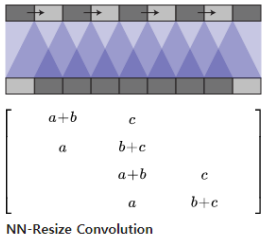
- Checkerboard artifact
  - Deconvolution (Transposed convolution) layer can easily have **“uneven overlap”**
  - Simple solution: use **“resize + conv”** or **“sub-pixel convolutional layer”**

Using deconvolution.
Heavy checkerboard artifacts.

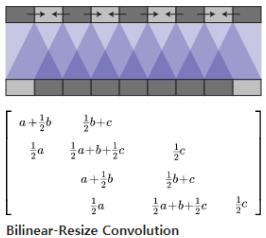
Using resize-convolution.
No checkerboard artifacts.



Deconvolution



NN-Resize Convolution



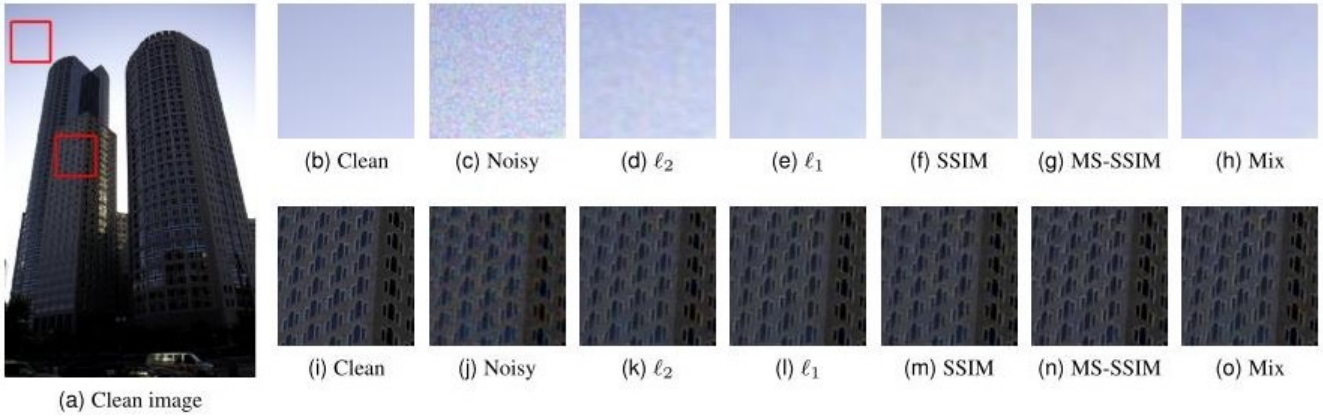
Bilinear-Resize Convolution

Reference: “Deconvolution and Checkerboard Artifacts”, distill blog(https://distill.pub/2016/deconv-checkerboard/)

Some Issues for Super Resolution

- Loss function
  - Propose a various loss function methods in Image Restoration task
  - Report the best result when using **mixed loss with MS-SSIM loss + l_1 loss**

$$\mathcal{L}^{\text{Mix}} = \alpha \cdot \mathcal{L}^{\text{MS-SSIM}} + (1-\alpha) \cdot G_{\sigma_G^M} \cdot \mathcal{L}^{\ell_1}, \qquad (14)$$



Reference: "Loss Functions for Image Restoration with Neural Networks", 2016 IEEE TCI

Some Issues for Super Resolution

- Loss function
  - Recent papers almost use l_1 loss

TABLE 1
Parameters comparison of CNN-based SR algorithms. GRL stands for Global residual learning, LRL means Local residual learning, MST is abbreviation of Multi-scale training.

Method	Input	Output	Blocks	Depth	Filters	Parameters	GRL	LRL	MST	Framework	Loss
SRCNN	bicubic	Direct		3	64	57k				Caffe	ℓ_2
FSRCNN	LR	Direct		8	56	12k				Caffe	ℓ_2
ESPCN	LR	Direct		3	64	20k				Theano	ℓ_2
SCN	bicubic	Prog.	✓	10	128	42k				Cuda-CovNet	ℓ_2
REDNet	bicubic	Direct		30	128	4,131k	✓	✓		Caffe	ℓ_2
VDSR	bicubic	Direct		20	64	665k	✓	✓		Caffe	ℓ_2
DRCN	bicubic	Direct		20	256	1,775k	✓			Caffe	ℓ_2
LapSRN	LR	Prog.	✓	24	64	812k	✓			MatConvNet	ℓ_1
DRRN	bicubic	Direct	✓	52	128	297k	✓	✓	✓	Caffe	ℓ_2
SRGAN	LR	Direct	✓	33	64	1500k				Theano/Lasagne	ℓ_2
DnCNN	bicubic	Direct		17	64	566k			✓	MatConvNet	ℓ_2
IRCNN	bicubic	Direct		7	64	188k			✓	MatConvNet	ℓ_2
FormResNet	bicubic	Direct	✓	20	64	671k	✓		✓	MatConvNet	ℓ_2, ℓ_{TV}
EDSR	LR	Direct	✓	65	256	43000k	✓	✓		Torch	ℓ_1
MDSR	LR	Direct	✓	162	64	8,000k	✓	✓	✓	Torch	ℓ_1
ZSSR	LR	Direct		8	64	225k	✓			Tensorflow	ℓ_1
MemNet	bicubic	Direct	✓	80	64	677k	✓	✓	✓	Caffe	ℓ_2
MS-LapSRN	LR	Prog.	✓	84	64	222k	✓	✓	✓	MatConvNet	ℓ_1
CMSC	bicubic	Direct	✓	35	64	1220k	✓	✓	✓	PyTorch	ℓ_2
CNF	bicubic	Direct		15	64	337K				Caffe	ℓ_2
IDN	LR	Direct	✓	31	64	796k	✓	✓		Caffe	ℓ_2, ℓ_1
BTSRN	LR	Direct	✓	22	64	410K	✓	✓		Tensorflow	ℓ_2
SelNet	LR	Direct		22	64	974K	✓	✓		MatConvNet	ℓ_2
CARN	LR	Direct	✓	32	64	1,592K	✓	✓	✓	PyTorch	ℓ_1
SRMD	LR	Direct		12	128	1482k				MatConvNet	ℓ_2
SRDenseNet	LR	Direct	✓	64	16-128	-	✓	✓		TensorFlow	ℓ_2
EnhanceNet	LR	Direct	✓	24	64	-		✓		TensorFlow	ℓ_2, ℓ_t, GAN
SRFeat	LR	Direct	✓	54	128	-	✓	✓		TensorFlow	ℓ_2, ℓ_p, GAN
SRRAM	LR	Direct	✓	64	64	1,090K	✓	✓	✓	Tensorflow	ℓ_1
D-DBPN	LR	Direct	✓	46	64	10000K	✓	✓		Caffe	ℓ_2
RDN	LR	Direct	✓	149	64	21900k	✓	✓		Torch	ℓ_1
ESRGAN	LR	Direct	✓	115	64	-	✓	✓		Pytorch	ℓ_1
RCAN	LR	Direct	✓	500	64	16,000k	✓	✓	✓	Pytorch	ℓ_1

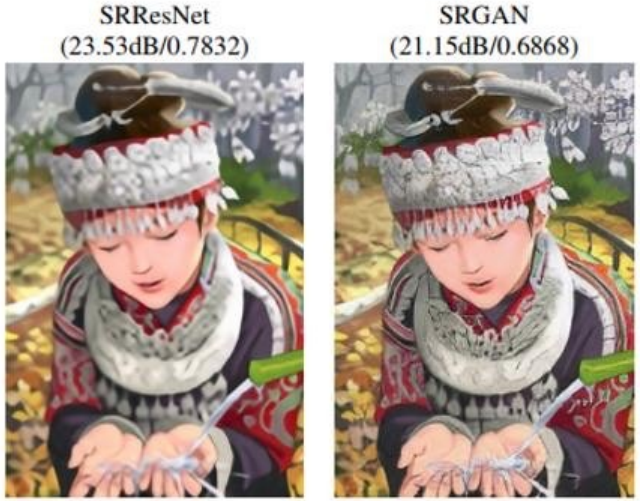
Reference: "A Deep Journey into Super-resolution: A survey", 2019 arXiv

Some Issues for Super Resolution

- Metric (Distortion measure)
  - Almost paper use distortion metric(PSNR and SSIM) as performance metric
  - But, high PSNR, SSIM do not guarantee good quality that looks good to human..

$$PSNR = 10 \cdot \log_{10}\left(\frac{MAX_I^2}{MSE}\right) = 20 \cdot \log_{10}\left(\frac{MAX_I}{\sqrt{MSE}}\right)$$

$$\mathrm{SSIM}(x,y) = \frac{(2\mu_x\mu_y + c_1)(2\sigma_{xy} + c_2)}{(\mu_x^2 + \mu_y^2 + c_1)(\sigma_x^2 + \sigma_y^2 + c_2)}$$

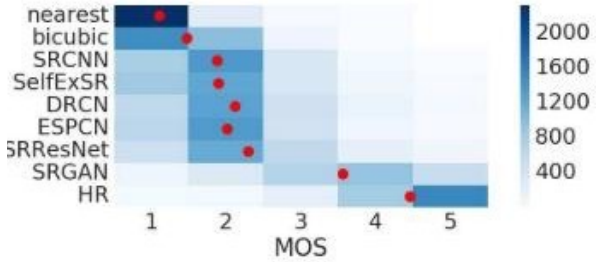


Reference: "Photo-Realistic Single Image Super-Resolution Using a Generative Adversarial Network", 2017 CVPR

Some Issues for Super Resolution

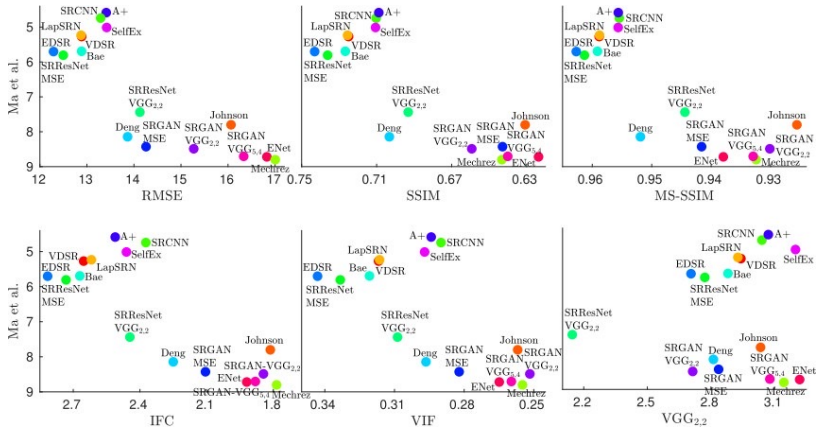
- Metric (Human opinion score)
  - But, high PSNR, SSIM do not guarantee good quality that looks good to human..
  - In SRGAN Paper, Mean Opinion Score(MOS) Testing is done for quantify perceptual quality
    - 26 raters, score from 1(bad) to 5(excellent)
    - The raters were calibrated on the NN(1) and HR(5) version of 20 images from BSD300 dataset

Set5	nearest	bicubic	SRCNN	SelfExSR	DRCN	ESPCN	**SRResNet**	**SRGAN**	HR
PSNR	26.26	28.43	30.07	30.33	31.52	30.76	**32.05**	29.40	∞
SSIM	0.7552	0.8211	0.8627	0.872	0.8938	0.8784	**0.9019**	0.8472	1
MOS	1.28	1.97	2.57	2.65	3.26	2.89	3.37	**3.58**	4.32
Set14									
PSNR	24.64	25.99	27.18	27.45	28.02	27.66	**28.49**	26.02	∞
SSIM	0.7100	0.7486	0.7861	0.7972	0.8074	0.8004	**0.8184**	0.7397	1
MOS	1.20	1.80	2.26	2.34	2.84	2.52	2.98	**3.72**	4.32
BSD100									
PSNR	25.02	25.94	26.68	26.83	27.21	27.02	**27.58**	25.16	∞
SSIM	0.6606	0.6935	0.7291	0.7387	0.7493	0.7442	**0.7620**	0.6688	1
MOS	1.11	1.47	1.87	1.89	2.12	2.01	2.29	**3.56**	4.46



Some Issues for Super Resolution

- Metric Paper (The Perception-Distortion Tradeoff, 2018 CVPR)
  - Analysis **distortion measure(PSNR, SSIM, etc.)** vs **human opinion score(Perceptual quality)**
  - Good supplementary video: https://www.youtube.com/watch?v=6Yid4dituqo (PR-12)



Reference: "The Perception-Distortion Tradeoff", 2018 CVPR

Deep Image Prior

Dmitry Ulyanov, Andrea Vedaldi, Victor Lempitsky

Partial content of this slide refers the presentation given by Dmitry Ulyanov regarding to the paper

Background and Motivation

- State-of-the-art ConvNets for image restoration and generation are almost invariably trained on large datasets of images. One may thus assume that their excellent performance is due to their ability to learn realistic image priors from a large number of example images.

- However, learning alone is insufficient to explain the good performance of deep networks.
  - Recent research has shown that generalization requires the **structure of the network** to “resonate” with the **structure of the data**.

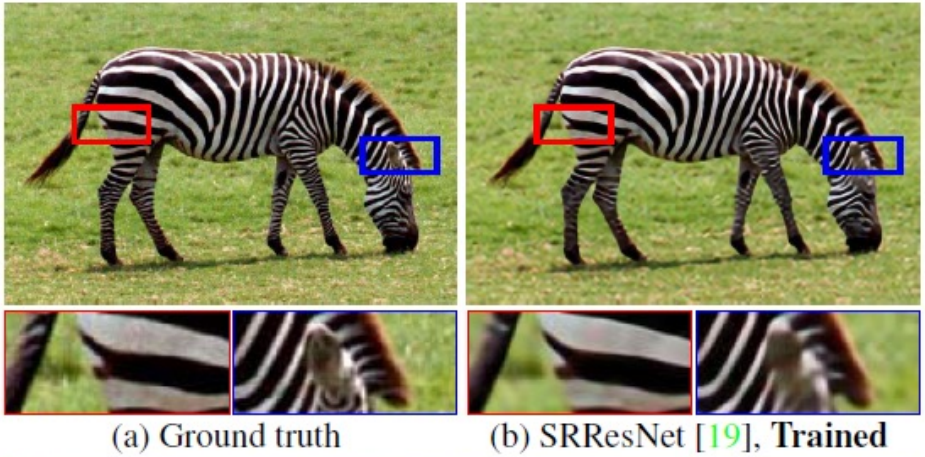
What did they do?

- In this paper, they show that, contrary to the belief that learning is necessary for building good image priors, a great deal of image statistics are captured by the structure of a convolutional image generator independent of learning.

- They cast reconstruction as a conditional image generation problem and show that the only information required to solve it **is contained in the single degraded input image** and the **handcrafted structure of the network** used for reconstruction.

- Instead of trying to beat the state-of-art neural networks, they try to show the structure of the network imposes strong prior.

Result



(a) Ground truth (b) SRResNet [19], **Trained**

(c) Bicubic, **Not trained** (d) Deep prior, **Not trained**

Image restoration - Method

x – Clean image
$\hat{x}$ – Corrupted/degraded image (observed)
x^*- Restored image

Degradation for denoising:

$$\hat{x} = x + \epsilon, \qquad \epsilon \in N(0, \sigma^2)$$

Restoration Model:

$$x^* = \arg\max_x p(x|\hat{x}) = \arg\max_x p(\hat{x}|x)p(x)$$

If there is no preference for prior, the prior will be a constant. Then,

$$x^* = \arg\max_x p(\hat{x}|x) = \arg\max_x N(\hat{x}; x, \sigma^2) = \hat{x}$$

=> the best estimation of the clean image is the corrupted image

x – Clean image

$\hat{x}$ – Corrupted image (observed)

x^*- Restored image

$$x^* = \arg\max_x p(x|\hat{x}) = \arg\max_x p(\hat{x}|x)p(x)$$
$$= \arg\min_x -logp(\hat{x}|x) - logp(x)$$

Expressed as energy minimization problem:

$$x^* = \arg\min_x E(x, \hat{x}) + R(x)$$

where $E(x, \hat{x})$ is a task-dependent data term, $R(x)$ is a regularizer

For example:

$$x^* = \arg\max_x p(\hat{x}|x) = \arg\max_x N(\hat{x}; x, \sigma^2) = \arg\min_x \|x - \hat{x}\|^2$$

Deep Image Prior

- $\hat{x}$ – Corrupted image (observed)

- Parametrization:
  Interpreting the neural network as a parametrization

Fixed input z

$$x \equiv f_\theta(z)$$

Convolutional network with parameter θ

- In particular, most of their experiments are performed using a U-Net type “hourglass” architecture(also known as “decoder-encoder”) with skip-connections, where *z* and *x* have the same spatial size.

Deep Image Prior step by step

$\hat{x}$ – Corrupted image (observed)

x^*- Restored image

1. initialize z

   For example fill it with uniform noise U(-1, 1)

2. Solve $\theta^* = \arg\min_{\theta} E(f_\theta(z); \hat{x}))$

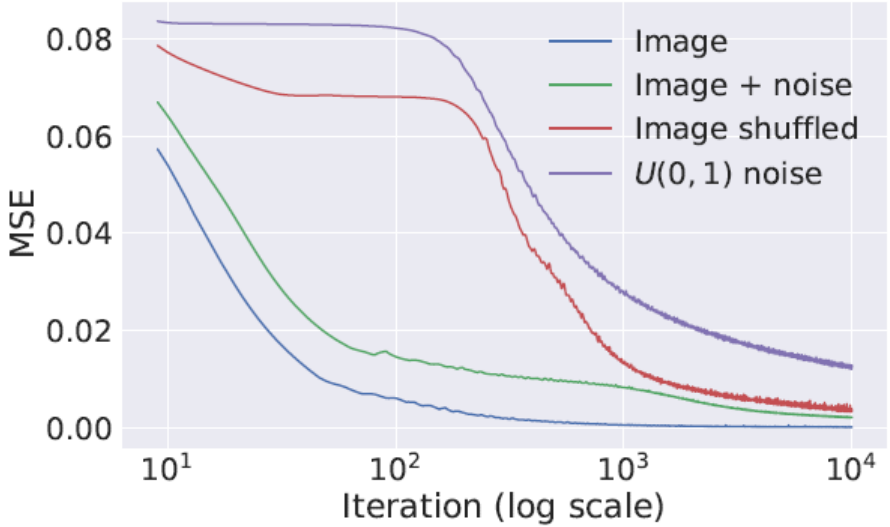
   With any favorite gradient-based method

$$\theta^{k+1} = \theta^k - \alpha \frac{\partial E(f_\theta(z); \hat{x}))}{\partial \theta};$$

3.Get the solution

$$x^* = f_{\theta^*}(z)$$

(a) Image (b) Image + noise (c) Image shuffled (d) $U(0, 1)$ noise



he network has high impedance to noise
nd low impedance to signal.

Therefore for most applications,
hey restrict the number of iterations in
ıe optimization process to a certain
umber of iterations.

Data term

x – Clean image
$\hat{x}$ – Corrupted image (observed)
m- Binary mask

Objective: $\theta^* = \arg\min_{\theta} E(f_\theta(z); \hat{x}))$

Denoising: $E(x, \hat{x}) = \|x - \hat{x}\|^2$ Needs early stopping!

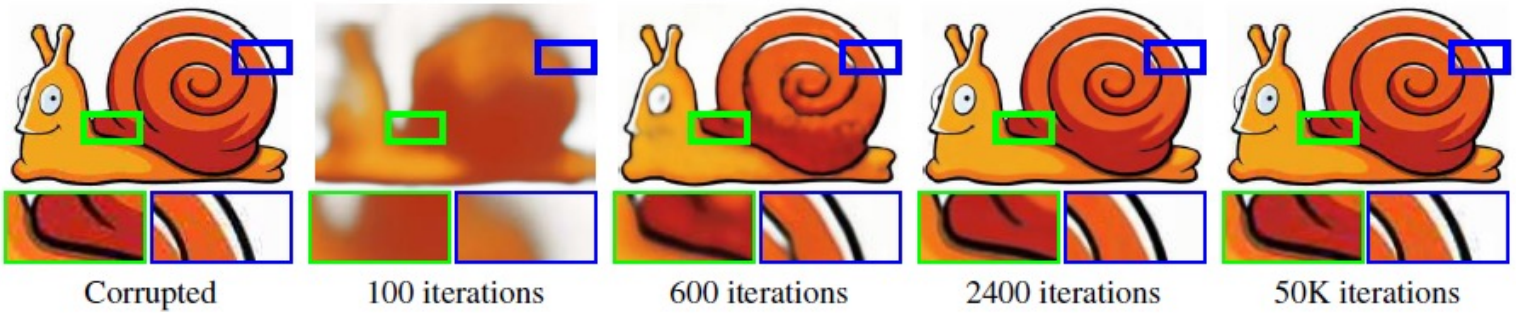
Inpainting: $E(x, \hat{x}) = \|(x - \hat{x}) \odot m\|^2$, where $\odot$ is Hadamard's product, m is binary mask

Super-resolution: $E(x, \hat{x}) = \|d(x) - \hat{x}\|^2$, where $d(\cdot)$ is a downsampling operator to resize the image

Feature-inv: $E(x, \hat{x}) = \|\phi(x) - \phi(\hat{x})\|^2$, where ϕ is the first several layers of a neural network trained to perform

Experiments

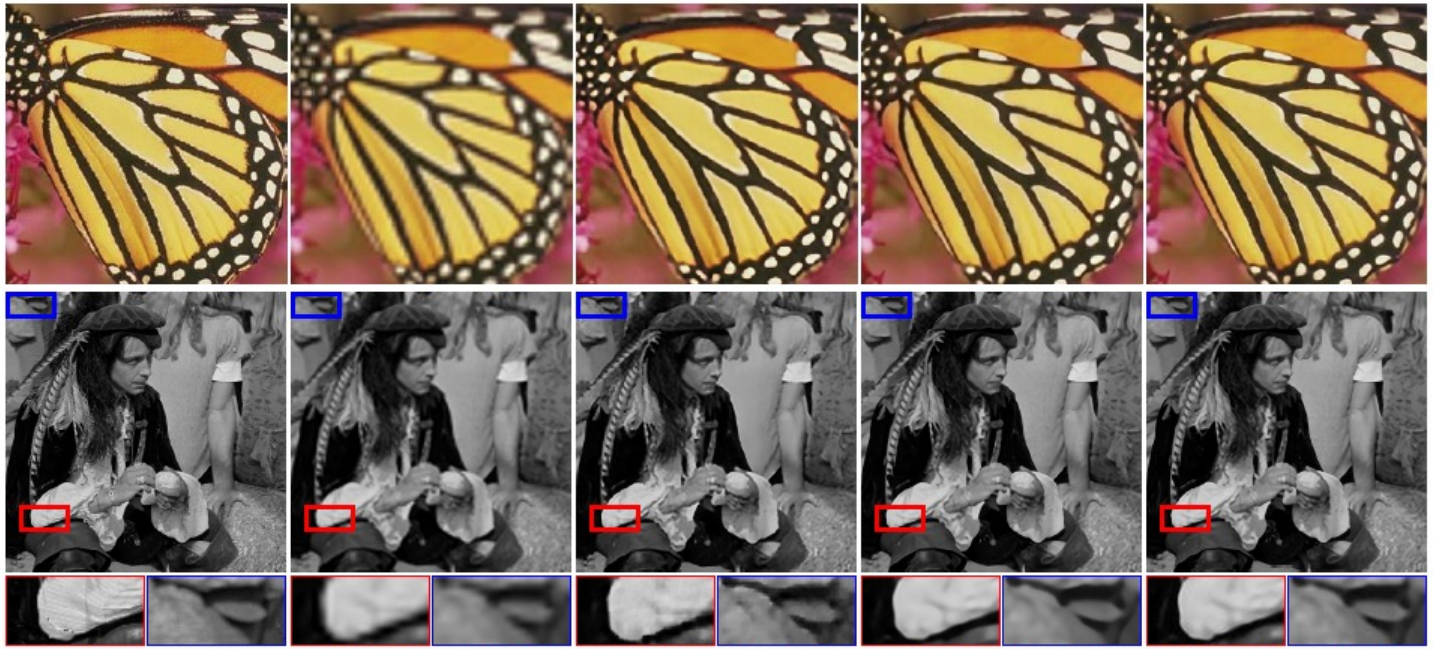
Denoising and generic reconstruction



Deep Image Prior approach can restore an image with a complex degradation (JPEG compression in this case). As the optimization process progresses, the deep image prior allows to recover most of the signal while getting rid of halos and blockiness (after 2400 iterations) before eventually overfitting to the input (at 50K iterations).

The deep image prior is successful at recovering both man-made and natural patterns.

Super-resolution

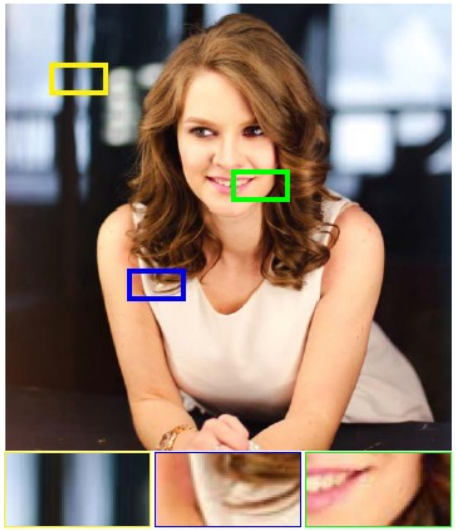


(a) Original image (b) Bicubic, **Not trained** (c) Ours, **Not trained** (d) LapSRN, **Trained** (e) SRResNet, **Trained**

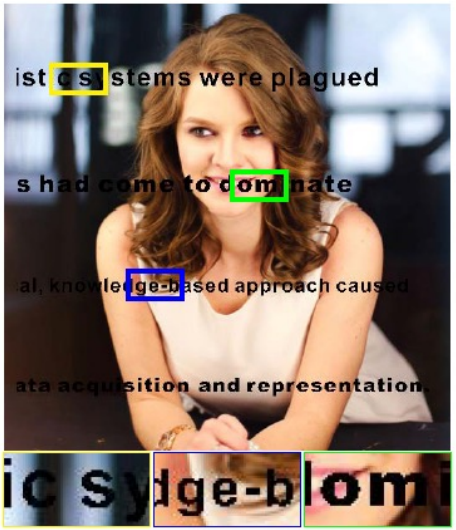
use a scaling factor of 4 to compare to other works
fix the number of optimization steps to be 2000 for every image

Inpainting

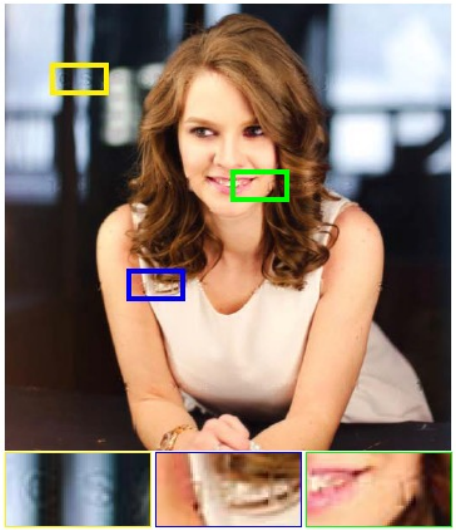
Text inpainting



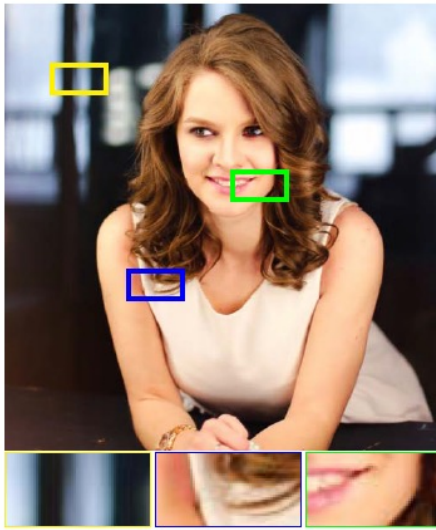
(a) Original image



(b) Corrupted image

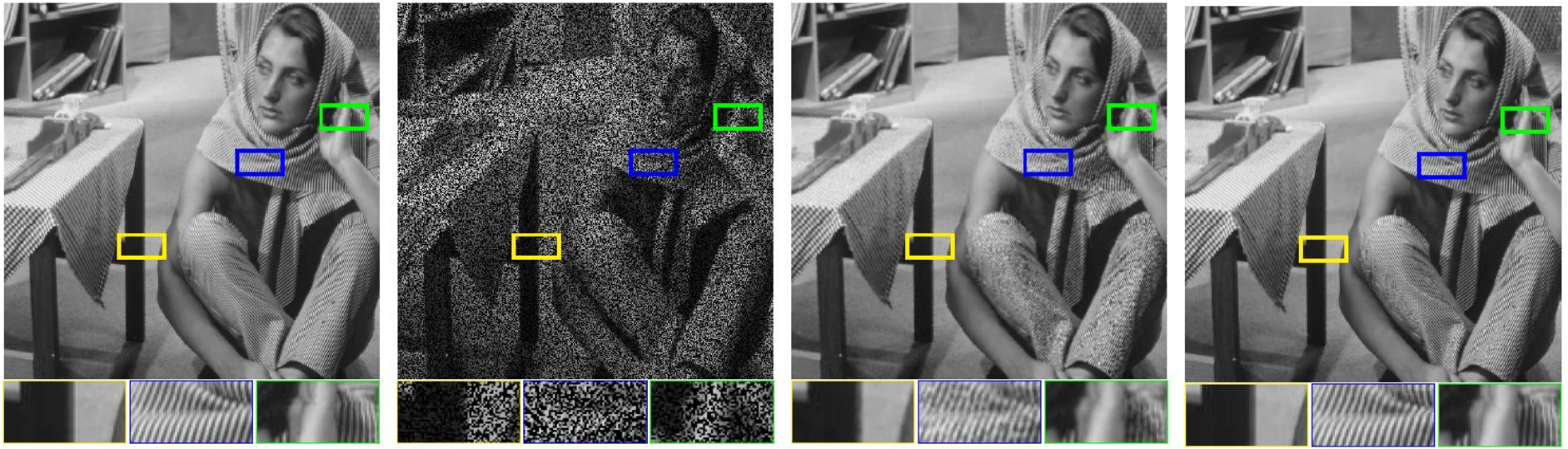


(c) Shepard networks [27]



(d) Deep Image Prior

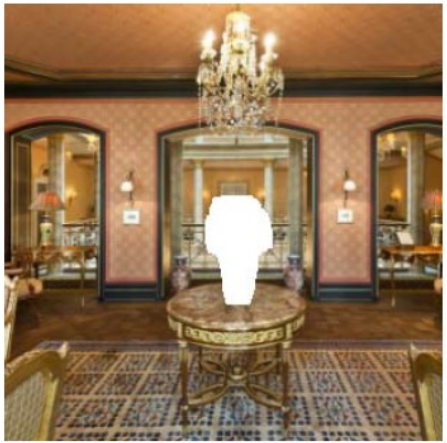
Image restoration



(e) Original image (f) Corrupted image (g) [25], PSNR = 28.1 (h) Deep Img. Prior, PSNR = 32.22

sampled to drop 50% of pixels at random
g is the result from comparison with Shepard networks

Inpainting of large holes



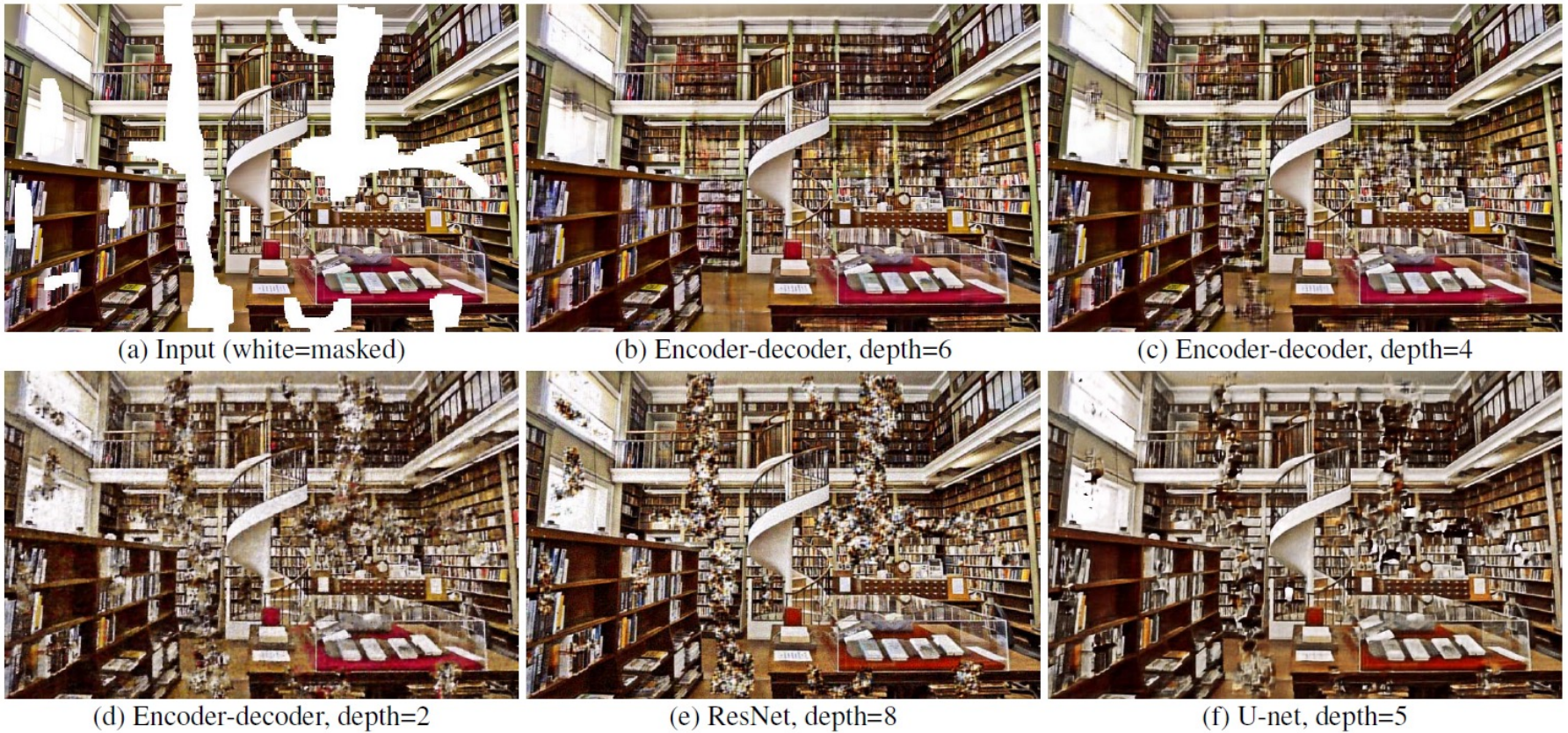
(a) Corrupted image

(b) Global-Local GAN [15]

(c) Ours, LR $= 0.01$

(d) Ours, LR $= 10^{-4}$

The deep image prior utilizes context of the image and interpolates the unknown region with textures from the known part. Such behaviour highlights the relation between the deep image prior and traditional self-similarity priors



(a) Input (white=masked) (b) Encoder-decoder, depth=6 (c) Encoder-decoder, depth=4

(d) Encoder-decoder, depth=2 (e) ResNet, depth=8 (f) U-net, depth=5

Inpainting using different depths and architectures.
The figure shows that much better inpainting results can be obtained by using deeper random networks. However, adding skip connections to ResNet in U-Net is highly detrimental.

Deep Video Prior: bridging image & video processing

Make image processing operators applicable to videos

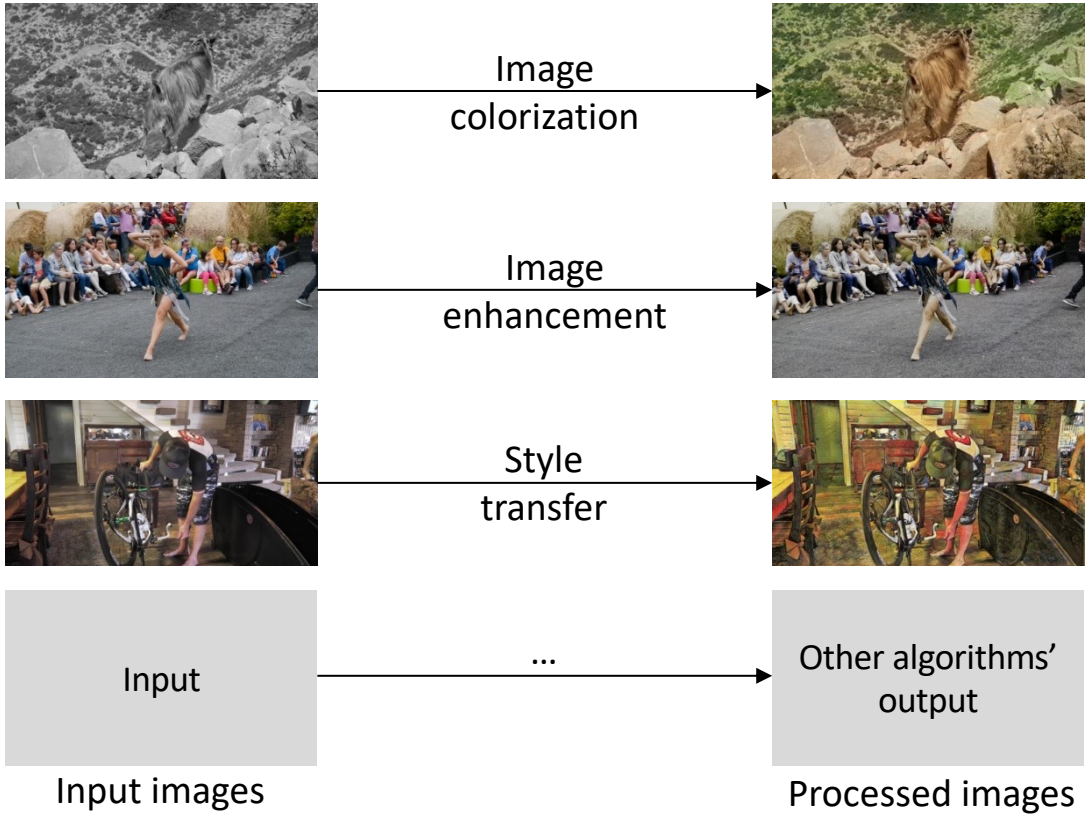
Input video

Image colorization

Processed video

Lei et al. Blind Video Temporal Consistency via Deep Video Prior. NeurIPS 2020

Tons of image algorithms



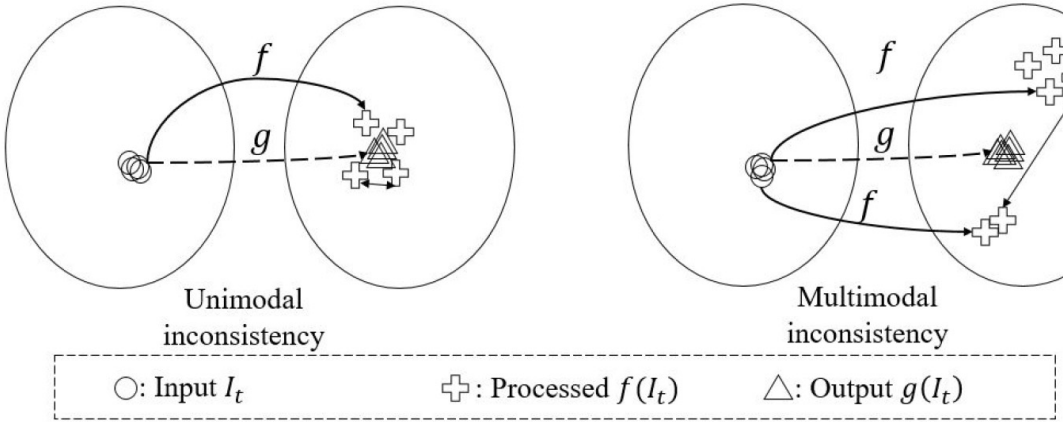
Improve the temporal consistency of processed videos

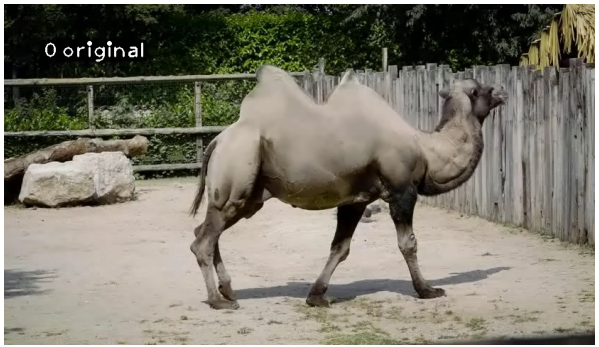
Processed video

Input video

Our algorithm

Output video





Unimodal inconsistency

Multimodal inconsistency

Deep Video Prior (DVP)

Deep video prior: the outputs of a CNN on two similar patches are expected to be similar at the early stage of training, and the same object in different video frames often has similar appearances

Image with noise

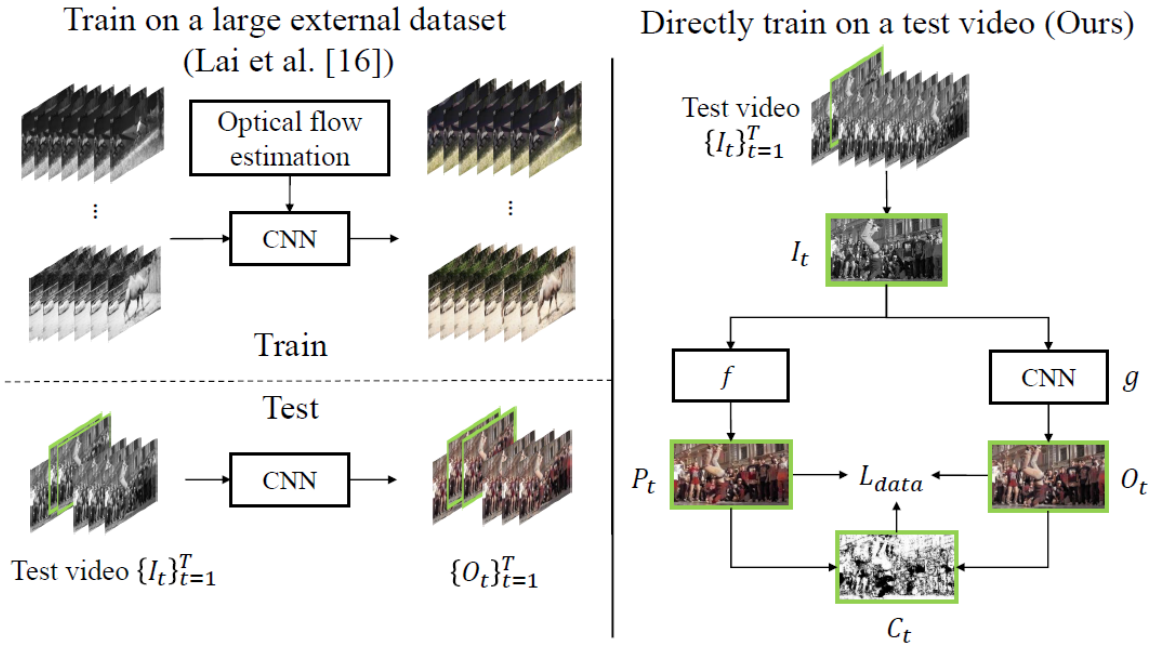
Noise-free image

Video with unimodal inconsistency

Deep Video Prior

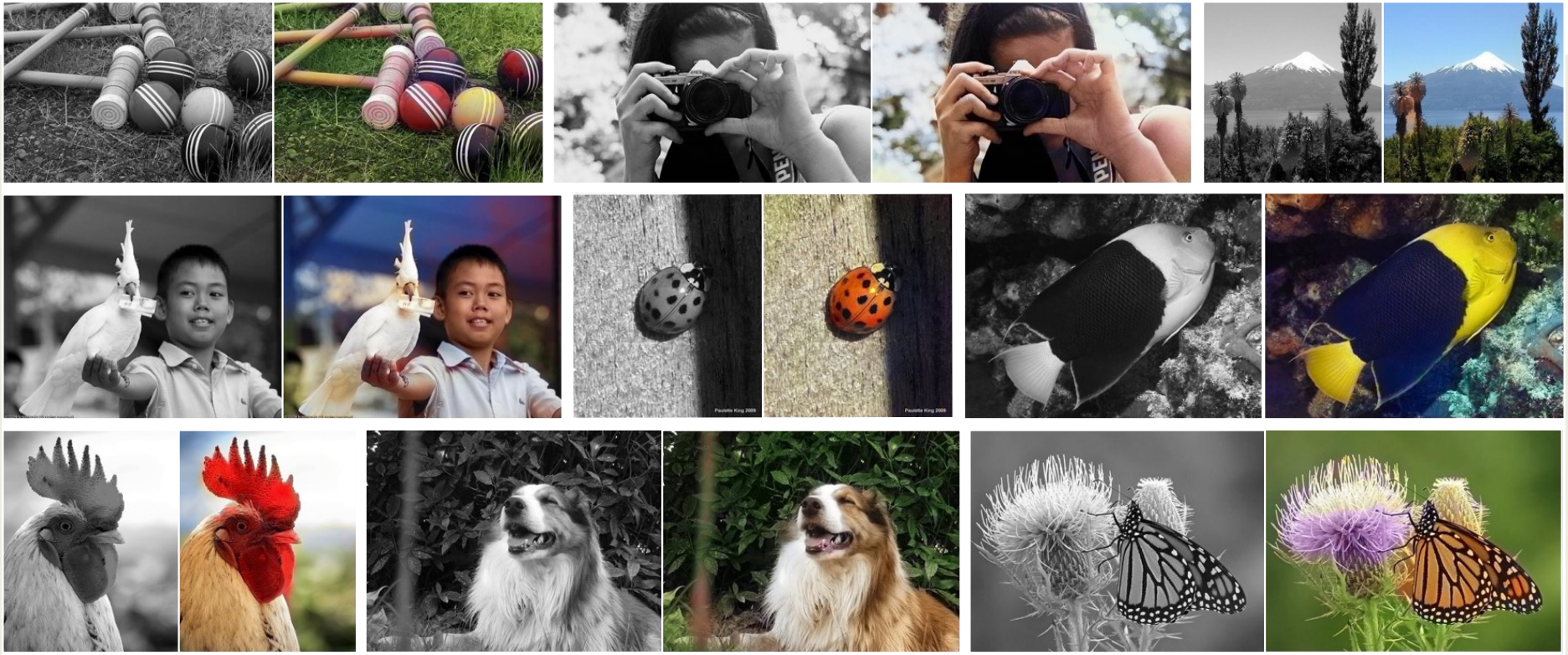
Temporal consistent video

Method


Train on a large external dataset
(Lai et al. [16])
Optical flow estimation
CNN
Train
Test
CNN
Test video $\{I_t\}_{t=1}^T$
$\{O_t\}_{t=1}^T$
Directly train on a test video (Ours)
Test video $\{I_t\}_{t=1}^T$
I_t
f
CNN
g
P_t
L_{data}
O_t
C_t

Results

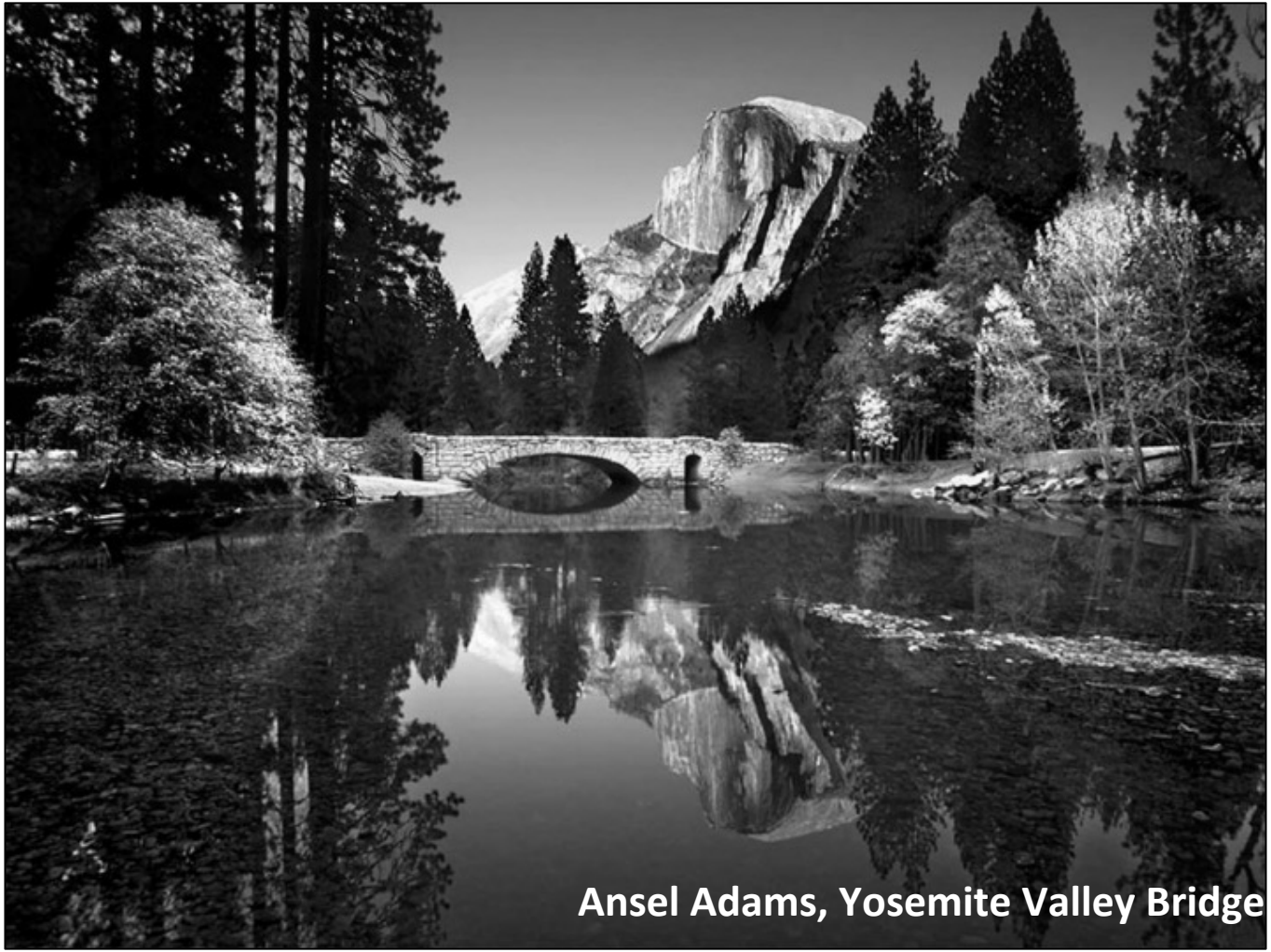
Results

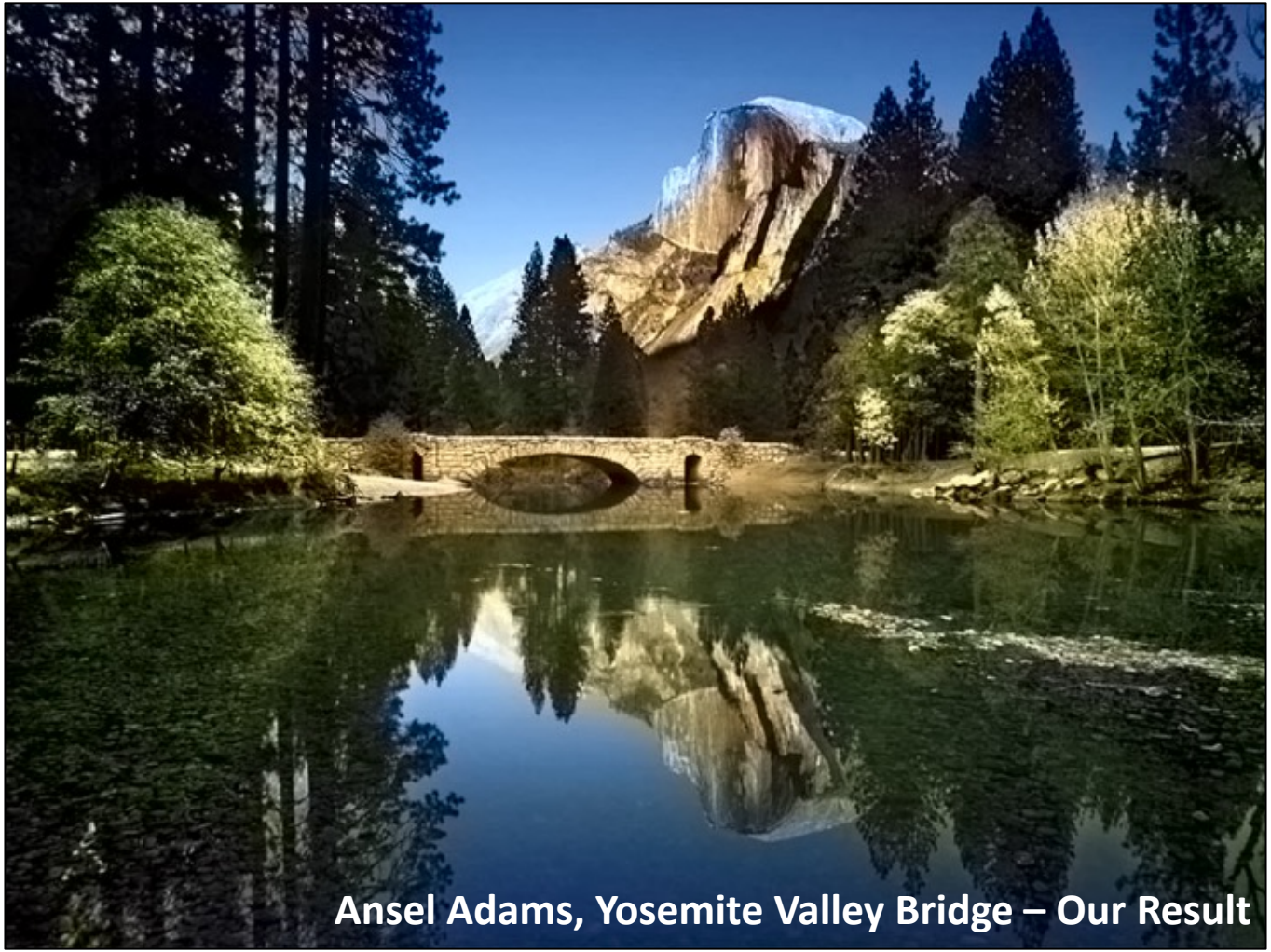


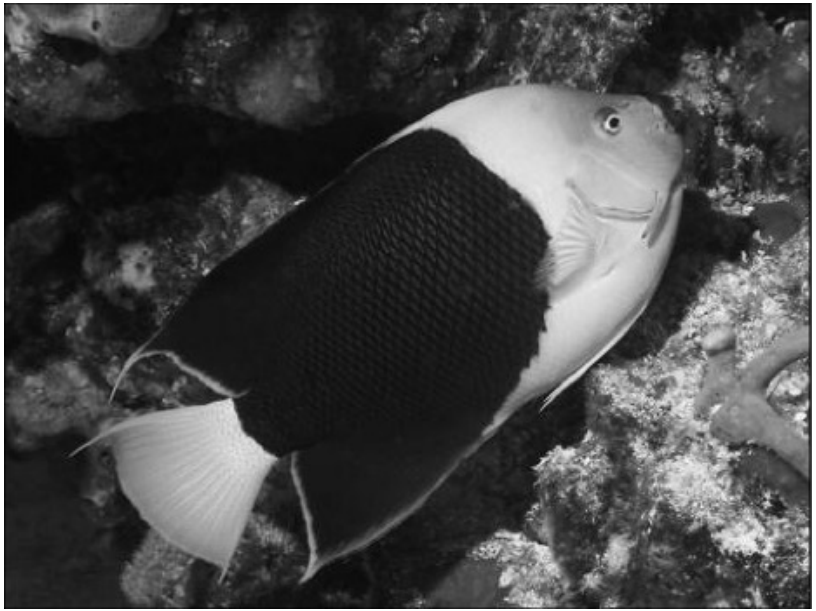
Colorful Image Colorization

Richard Zhang, Phillip Isola, Alexei (Alyosha) Efros

richzhang.github.io/colorization


Ansel Adams, Yosemite Valley Bridge


Ansel Adams, Yosemite Valley Bridge – Our Result



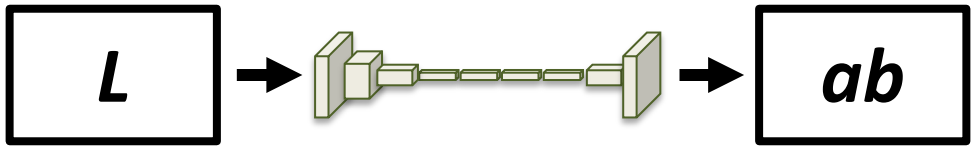
$\mathcal{F}$

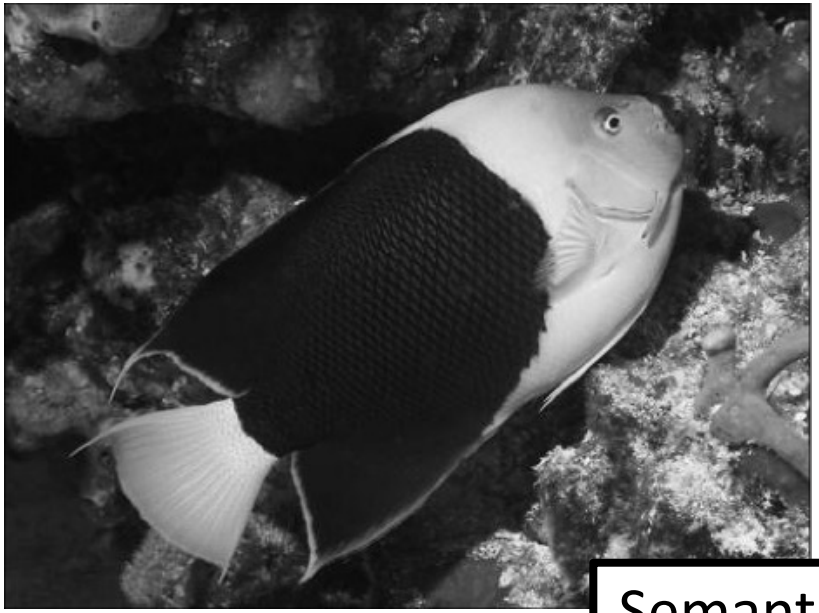
Grayscale image: *L* channel

$\mathbf{X} \in \mathbb{R}^{H \times W \times 1}$

Color information: *ab* channels

$\hat{\mathbf{Y}} \in \mathbb{R}^{H \times W \times 2}$



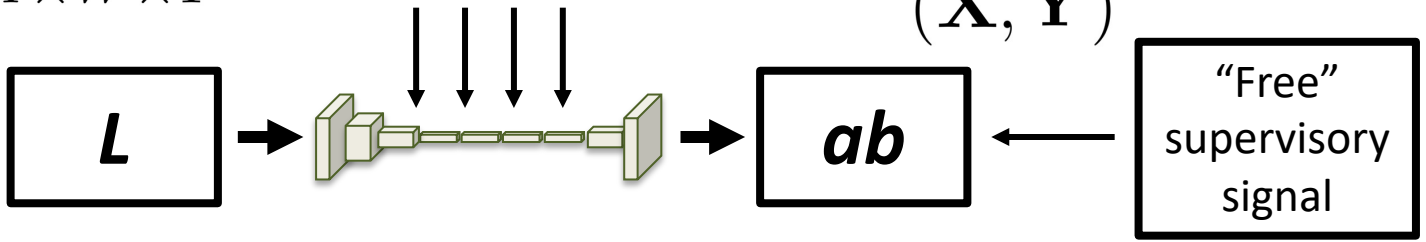


$\mathcal{F}$

Semantics? Higher-level abstraction?

Grayscale image: *L* ch… ncatenate (L,ab)

$\mathbf{X} \in \mathbb{R}^{H \times W \times 1}$ $(\mathbf{X}, \widehat{\mathbf{Y}})$



Inherent Ambiguity

Grayscale

Inherent Ambiguity

Our Output

Ground Truth

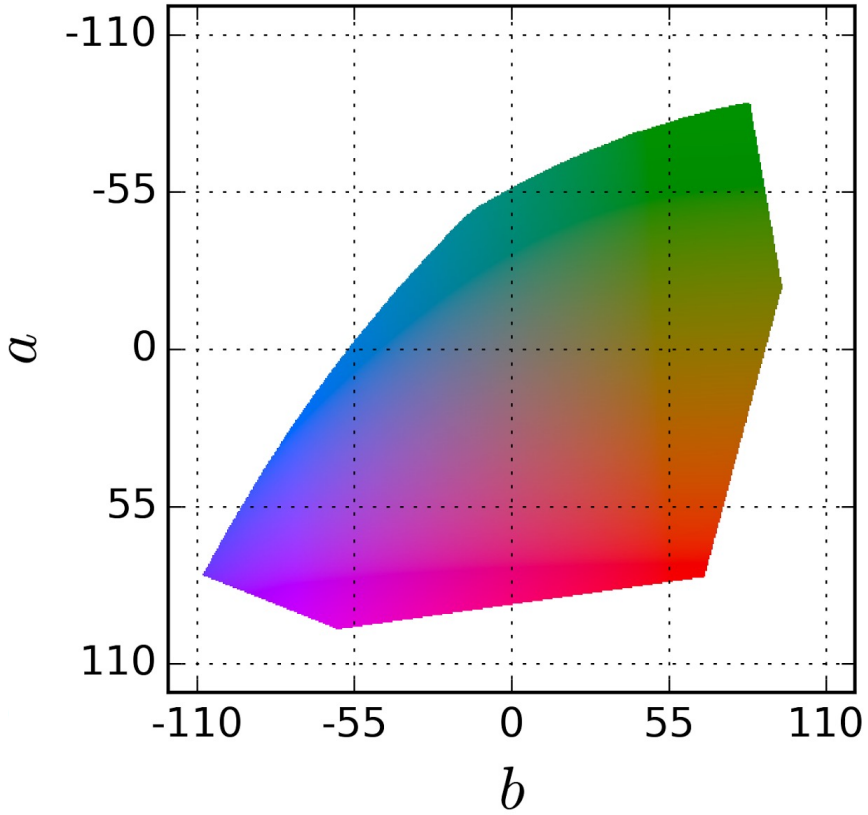
Better Loss Function

- Regression with L2 loss inadequate

$$\mathrm{L}_2(\widehat{\mathbf{Y}}, \mathbf{Y}) = \frac{1}{2} \sum_{h,w} \|\mathbf{Y}_{h,w} - \widehat{\mathbf{Y}}_{h,w}\|_2^2$$

Colors in *ab* space

(continuous)



Better Loss Function

- Regression with L2 loss inadequate

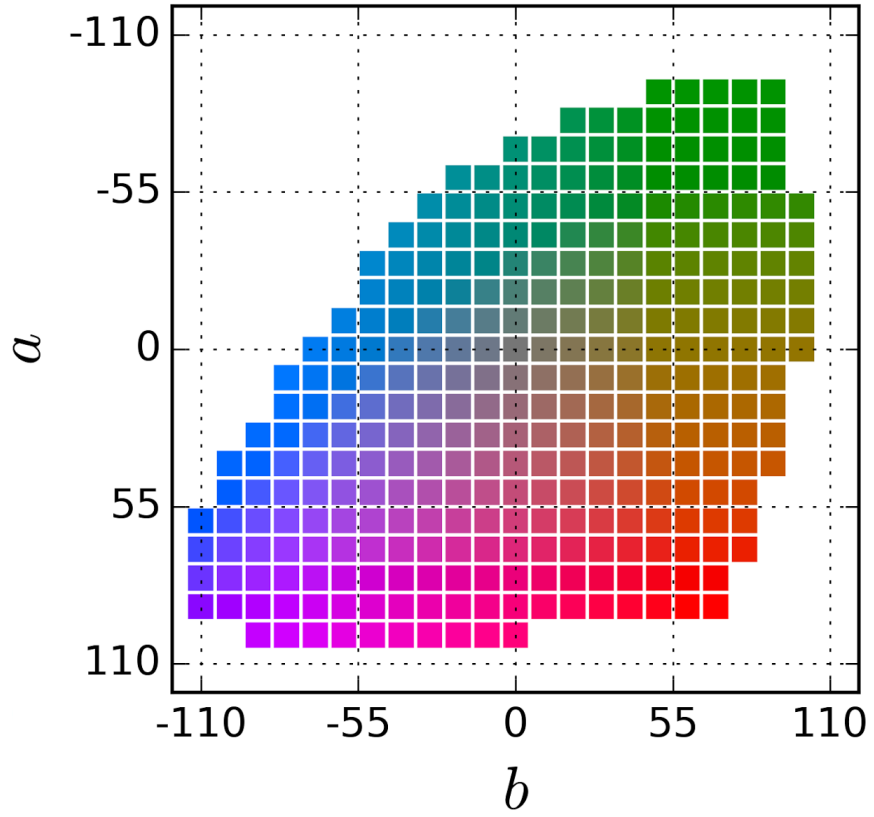
$$\mathrm{L}_2(\widehat{\mathbf{Y}}, \mathbf{Y}) = \frac{1}{2}\sum_{h,w} \|\mathbf{Y}_{h,w} - \widehat{\mathbf{Y}}_{h,w}\|_2^2$$

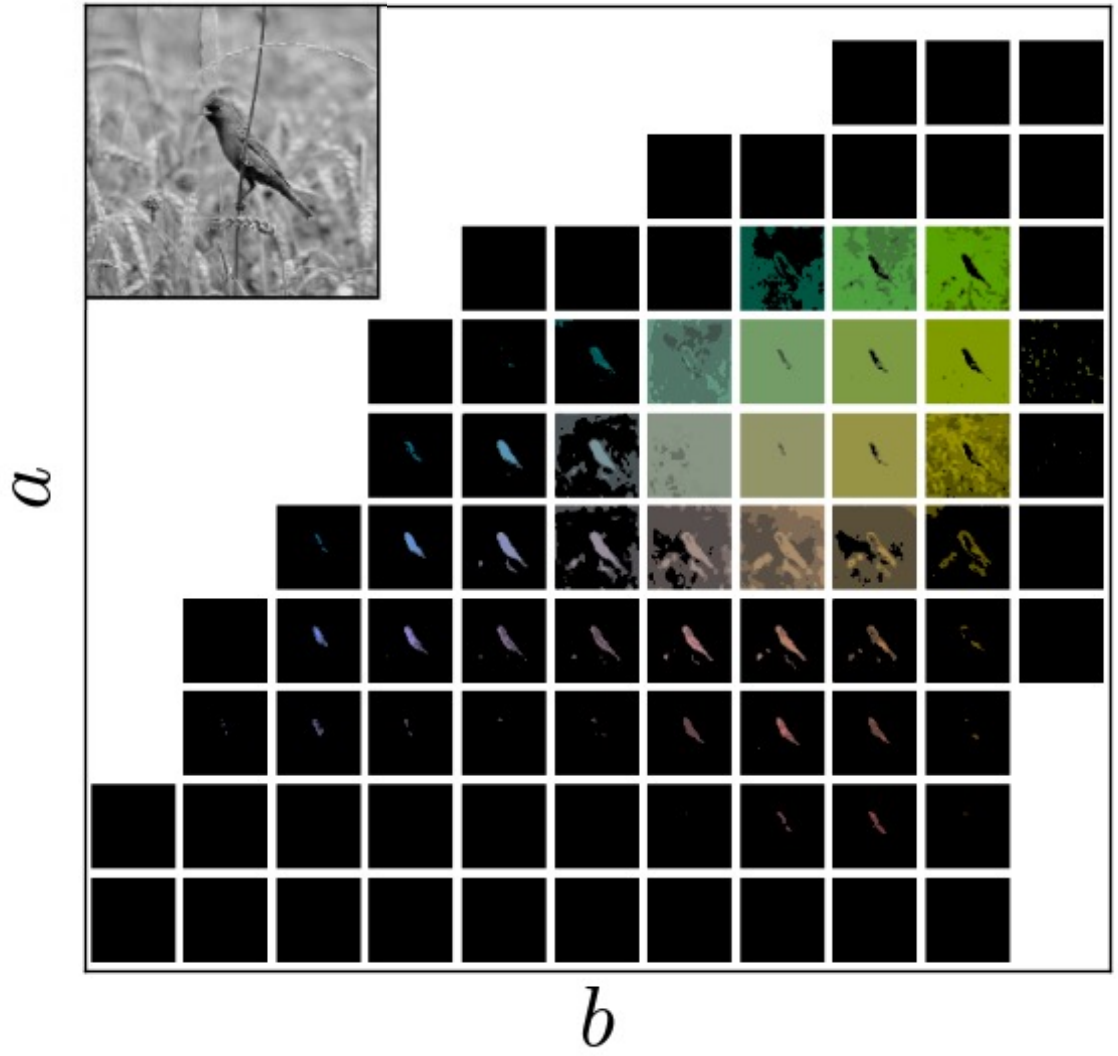
- Use **multinomial classification**

$$\mathrm{L}(\widehat{\mathbf{Z}}, \mathbf{Z}) = -\frac{1}{HW}\sum_{h,w}\sum_{q} \mathbf{Z}_{h,w,q} \log(\widehat{\mathbf{Z}}_{h,w,q})$$

Colors in *ab* space

(discrete)




a
b

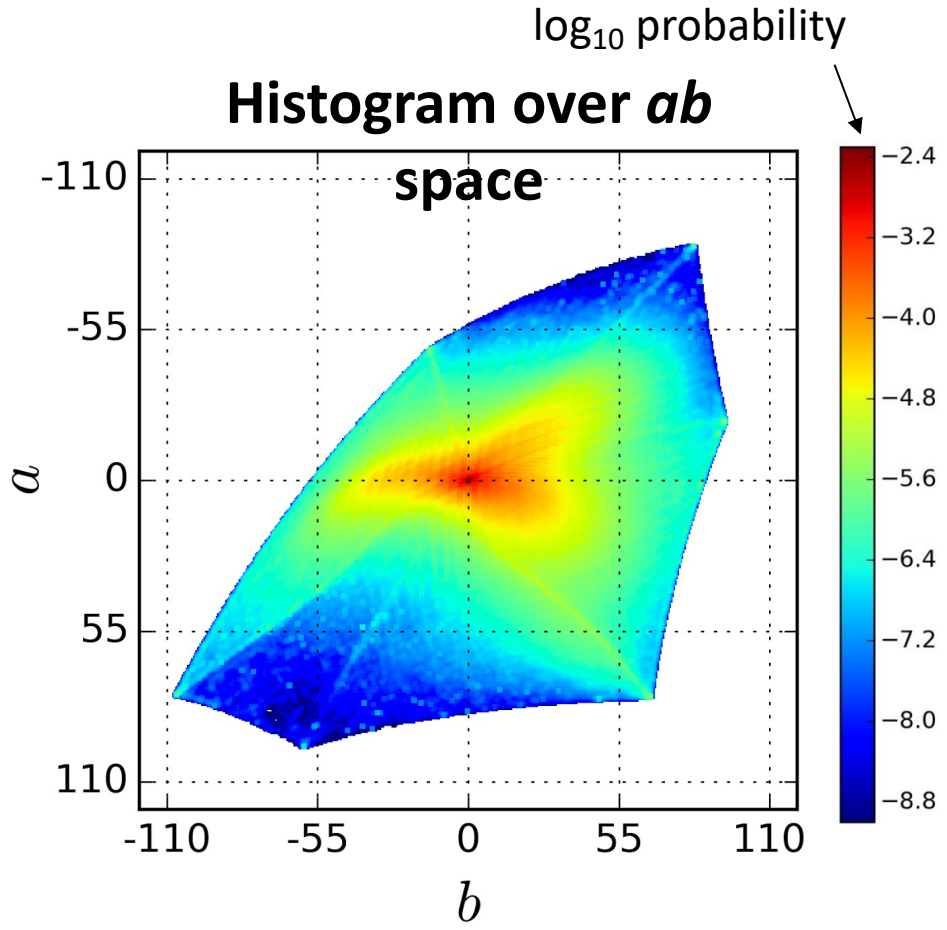
Better Loss Function

- Regression with L2 loss inadequate

$$\mathrm{L}_2(\widehat{\mathbf{Y}}, \mathbf{Y}) = \frac{1}{2} \sum_{h,w} \|\mathbf{Y}_{h,w} - \widehat{\mathbf{Y}}_{h,w}\|_2^2$$

- Use **multinomial classification**

$$\mathrm{L}(\widehat{\mathbf{Z}}, \mathbf{Z}) = -\frac{1}{HW} \sum_{h,w} \sum_{q} \mathbf{Z}_{h,w,q} \log(\widehat{\mathbf{Z}}_{h,w,q})$$



Better Loss Function

- Regression with L2 loss inadequate

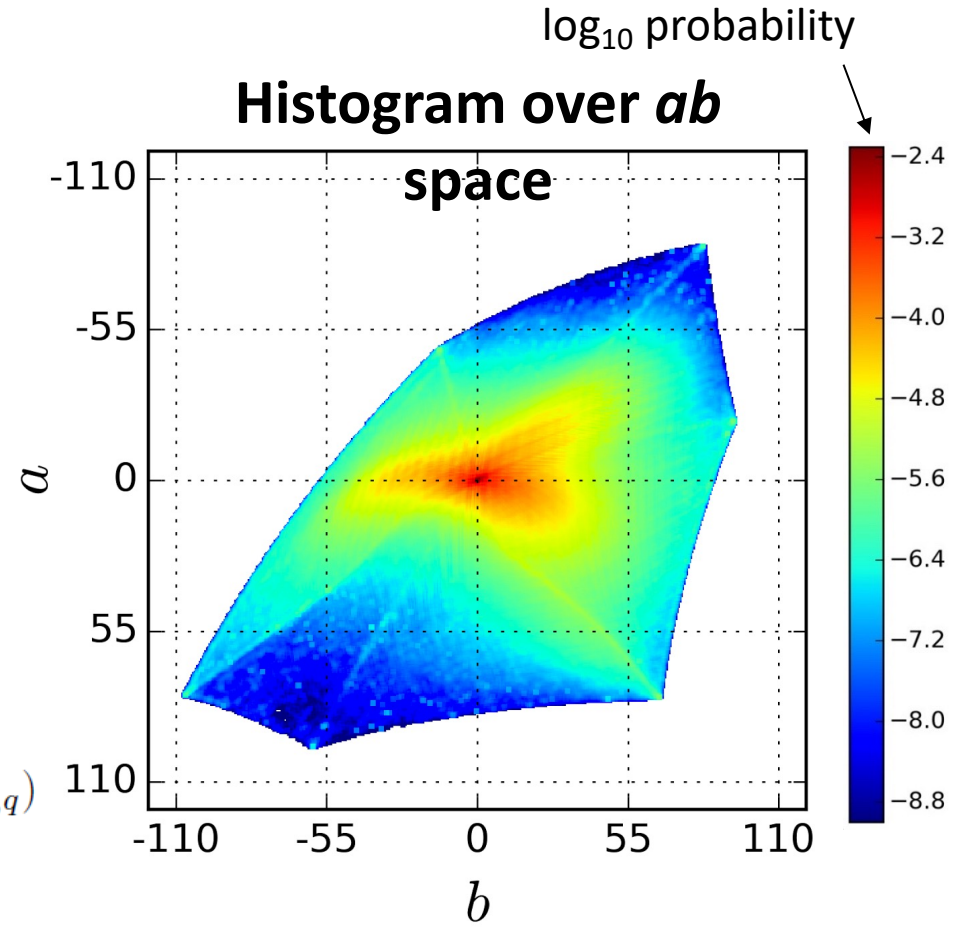
$$\mathrm{L}_2(\widehat{\mathbf{Y}}, \mathbf{Y}) = \frac{1}{2} \sum_{h,w} \|\mathbf{Y}_{h,w} - \widehat{\mathbf{Y}}_{h,w}\|_2^2$$

- Use **multinomial classification**

$$\mathrm{L}(\widehat{\mathbf{Z}}, \mathbf{Z}) = -\frac{1}{HW} \sum_{h,w} \sum_{q} \mathbf{Z}_{h,w,q} \log(\widehat{\mathbf{Z}}_{h,w,q})$$

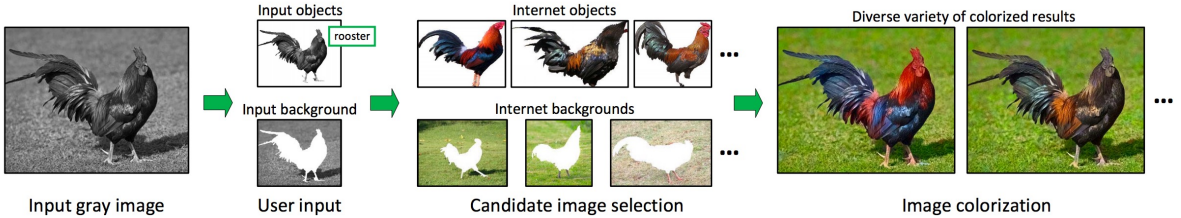
- **Class rebalancing** to encourage learning of rare colors

$$\mathrm{L}(\widehat{\mathbf{Z}}, \mathbf{Z}) = -\frac{1}{HW} \sum_{h,w} v(\mathbf{Z}_{h,w}) \sum_{q} \mathbf{Z}_{h,w,q} \log(\widehat{\mathbf{Z}}_{h,w,q})$$



Non-parametric

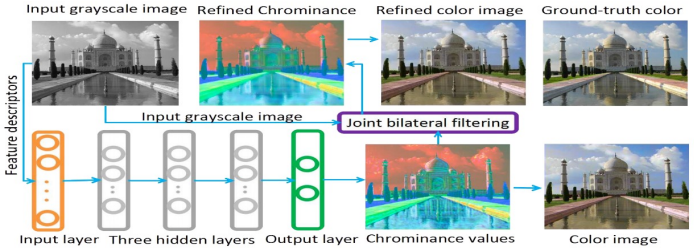
Hertzmann et al. In SIGGRAPH, 2001.
Welsh et al. In TOG, 2002.
Irony et al. In Eurographics, 2005.
Liu et al. In TOG, 2008.
Chia et al. In ACM 2011.
Gupta et al. In ACM, 2012.



Parametric

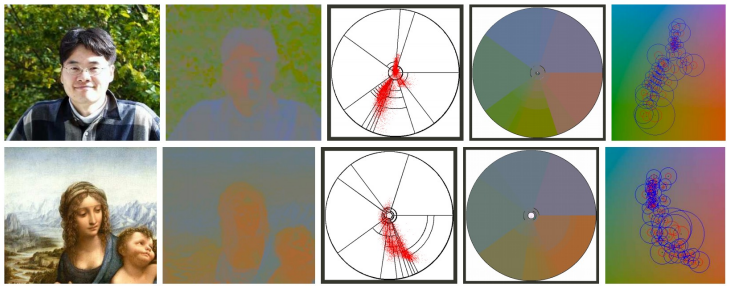
Hand-engineered Features

L2 Regression



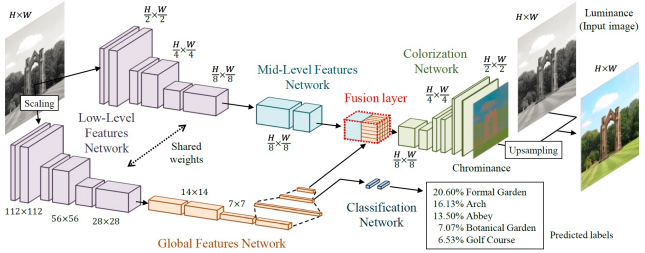
Deshpande et al. Cheng et al. In ICCV 2015.

Classification

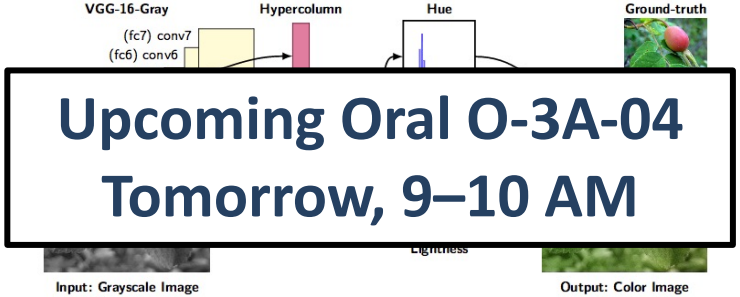


Charpiat et al. In ECCV 2008.

Deep Networks



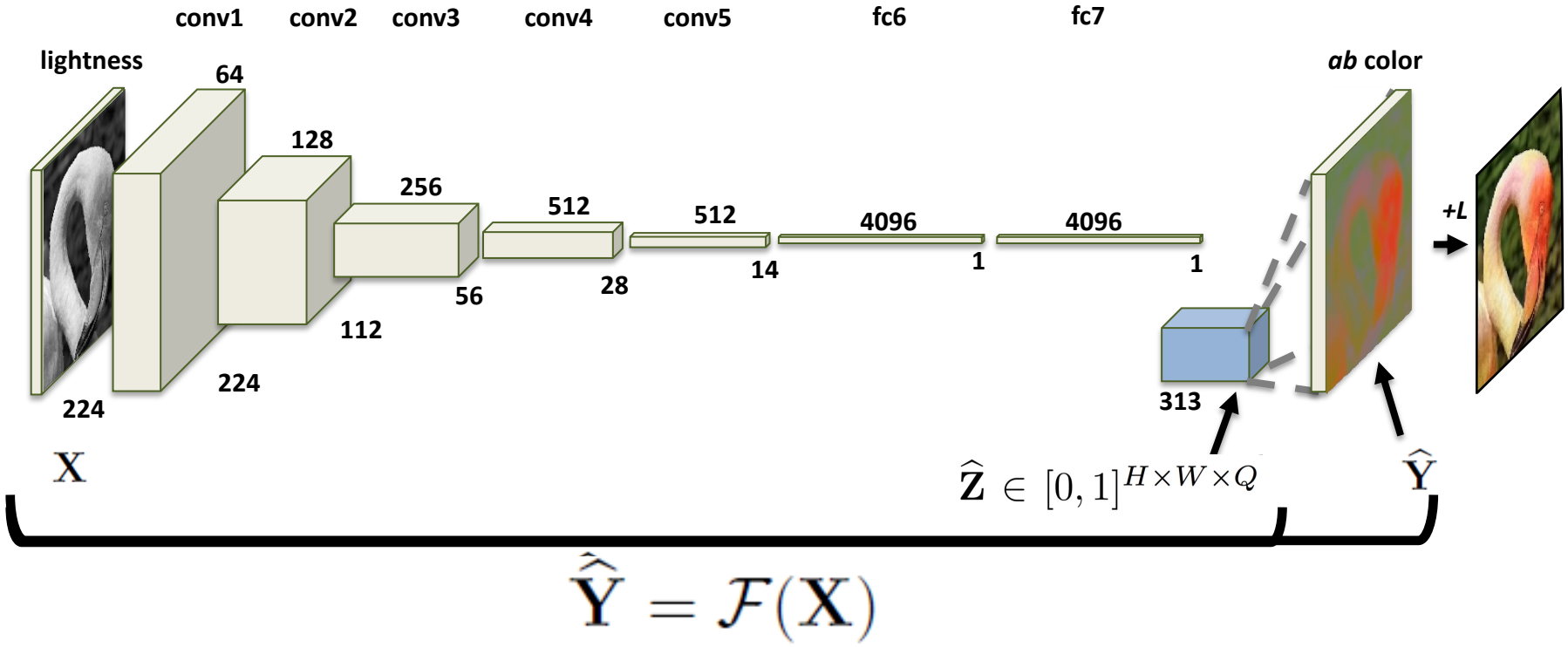
Dahl. Jan 2016. Iizuka et al. In SIGGRAPH, 2016.



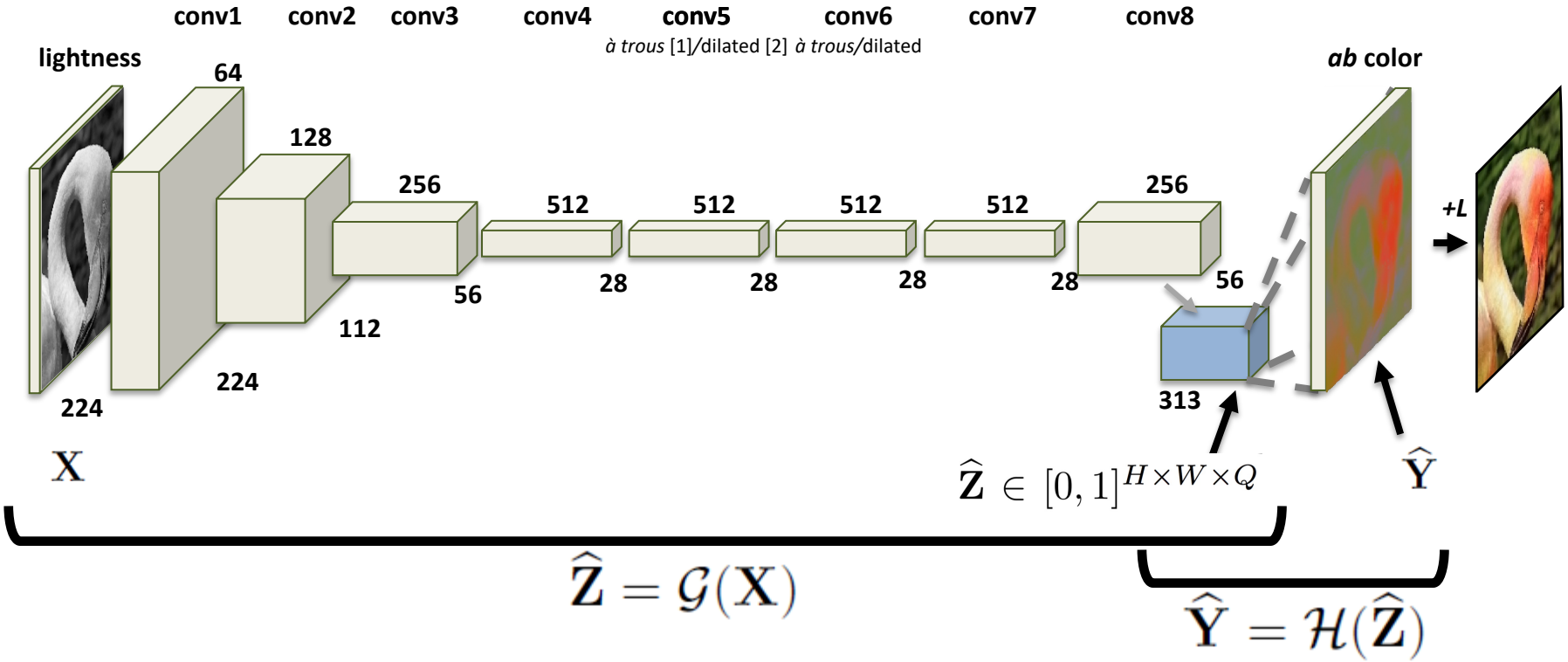
Upcoming Oral O-3A-04
Tomorrow, 9–10 AM

Larsson et al. In ECCV 2016. [Concurrent]

Network Architecture



Network Architecture



[1] Chen *et al.* In arXiv, 2016.
[2] Yu and Koltun. In ICLR,

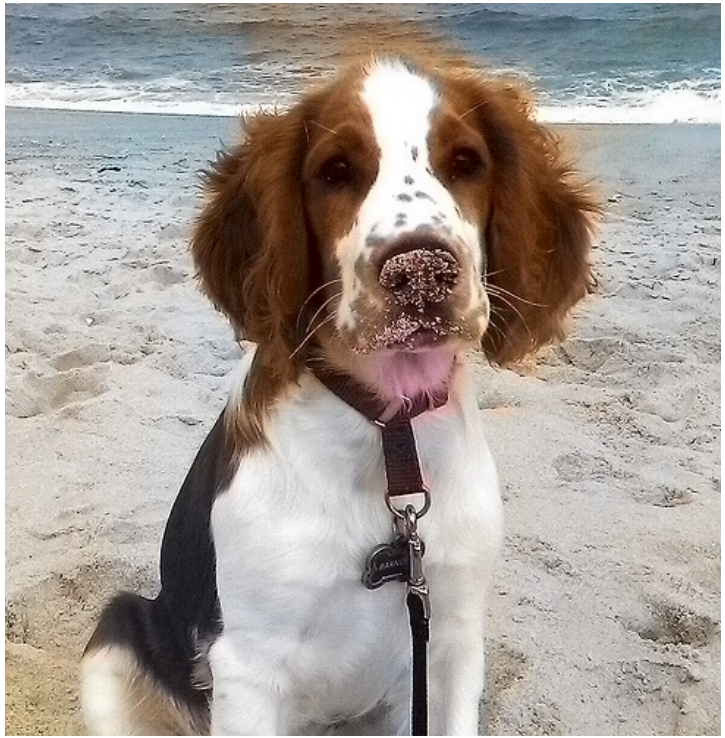
GT

L2 Regression

Class w/ Rebalancing

Failure Cases

Biases



Evaluation

	Visual Quality
Quantitative	Per-pixel accuracy

Evaluation

	Visual Quality	Representation Learning
Quantitative	Per-pixel accuracy Perceptual realism Semantic interpretability	Task generalization ImageNet classification Task & dataset generalization PASCAL classification, detection, segmentation
Qualitative	Low-level stimuli Legacy grayscale photos	Hidden unit activations

Evaluation

	Visual Quality	Representation Learning
Quantitative	Per-pixel accuracy **Perceptual realism** Semantic interpretability	Task generalization ImageNet classification Task & dataset generalization PASCAL classification, detection, segmentation
Qualitative	Low-level stimuli Legacy grayscale photos	Hidden unit activations

Perceptual Realism / Amazon Mechanical Turk Test

clap if “fake”

clap if “fake”

Fake, 0% fooled

clap if “fake”

clap if “fake”

Fake, 55% fooled

clap if “fake”

clap if “fake”

Fake, 58% fooled

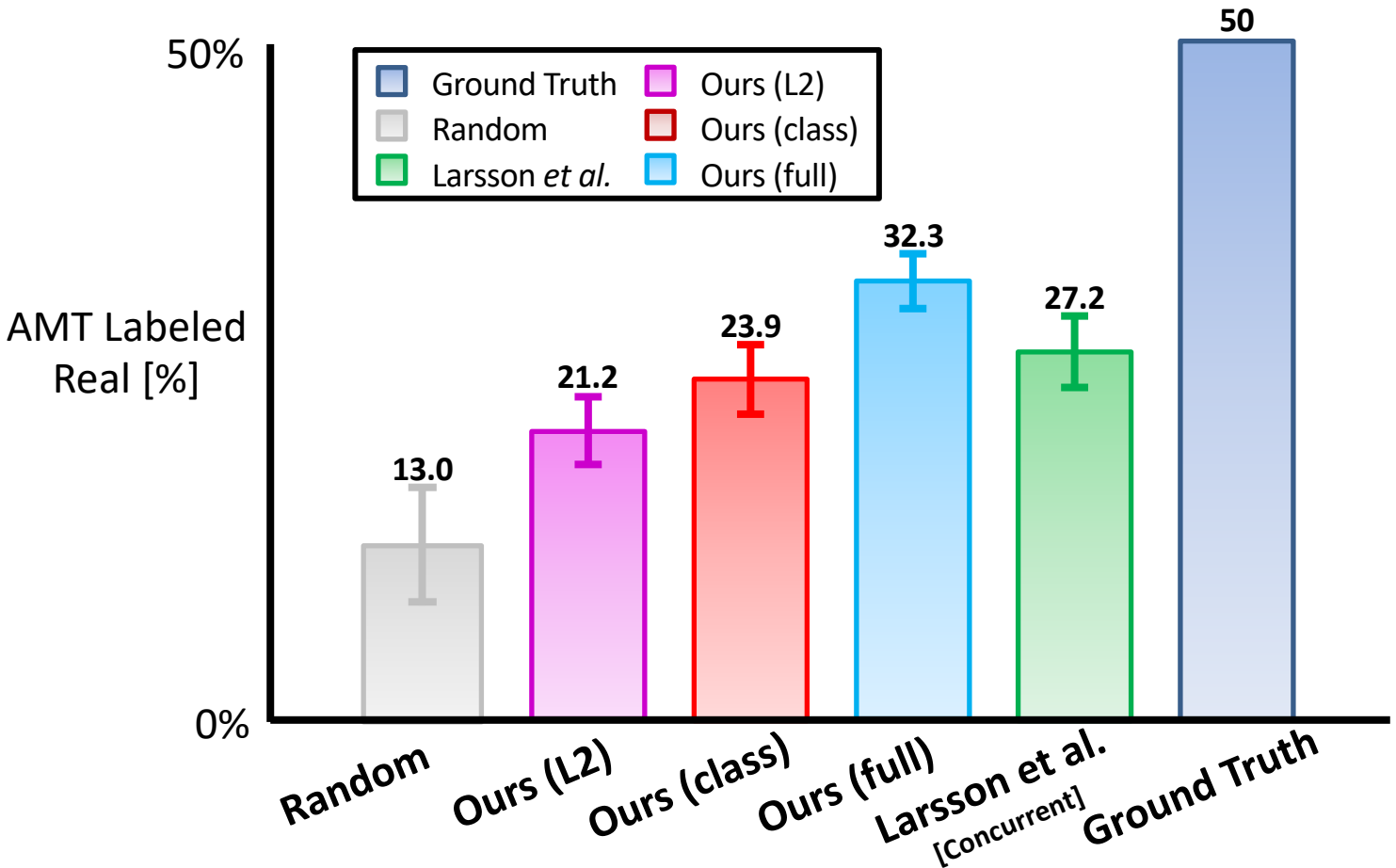
from Reddit /u/SherySantucci

Recolorized by Reddit ColorizeBot

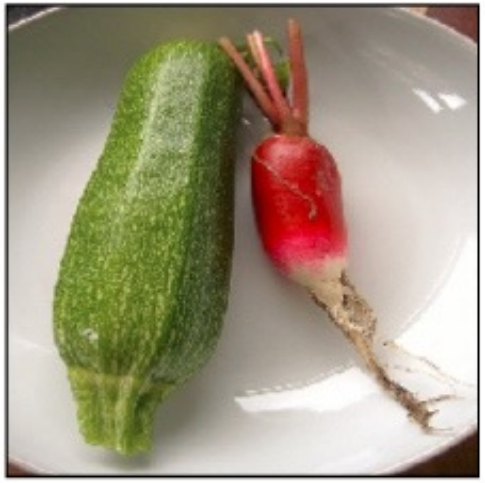
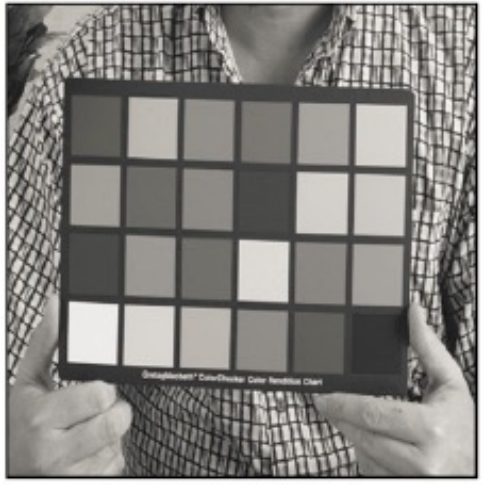
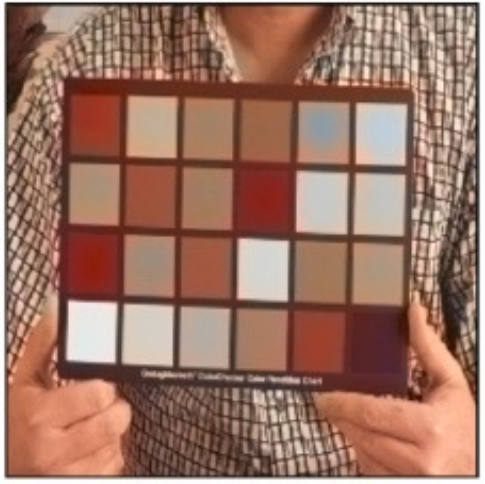
Photo taken by Reddit /u/Timteroo, Mural from street artist Eduardo Kobra

Recolorized by Reddit ColorizeBot

Perceptual Realism Test



1600 images tested per algorithm

Input	Ground Truth	Output
		
		

The End