Image2Image applications

Roberto Paredes

Centro de Investigación
Pattern Recognition and Human Language Technologies
Universidad Politécnica de Valencia

Index

- Introduction
- Image Colorization
- Image Inpainting
- Image Super-resolution

Introduction

- Image classification
- Object detection
- Image-to-image, target is a map

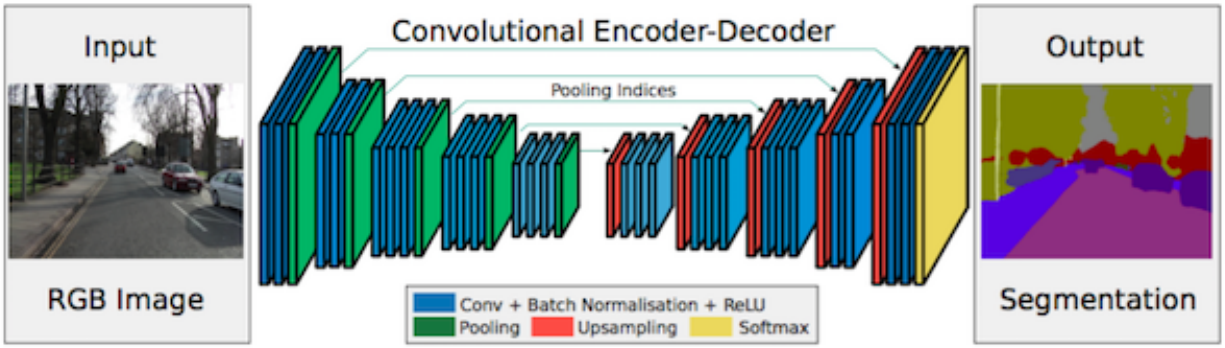


Image Colorization

https://www.youtube.com/watch?v=MfaTOXxA8dM

Image Colorization

- Gray map to color map
- Training data is all around
- Ambiguity problem
- Methods have to capture semantics

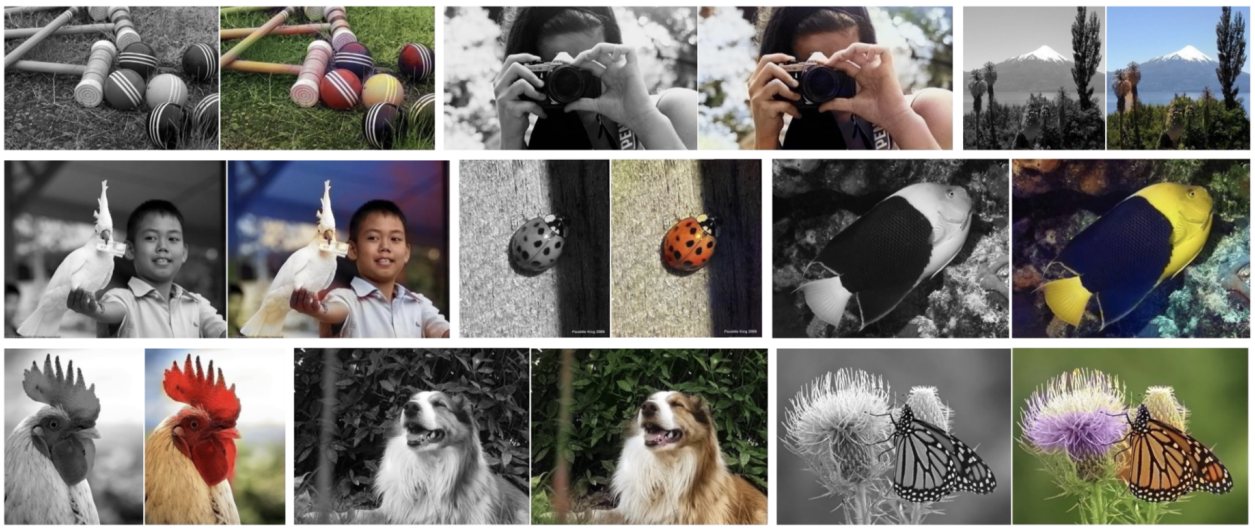


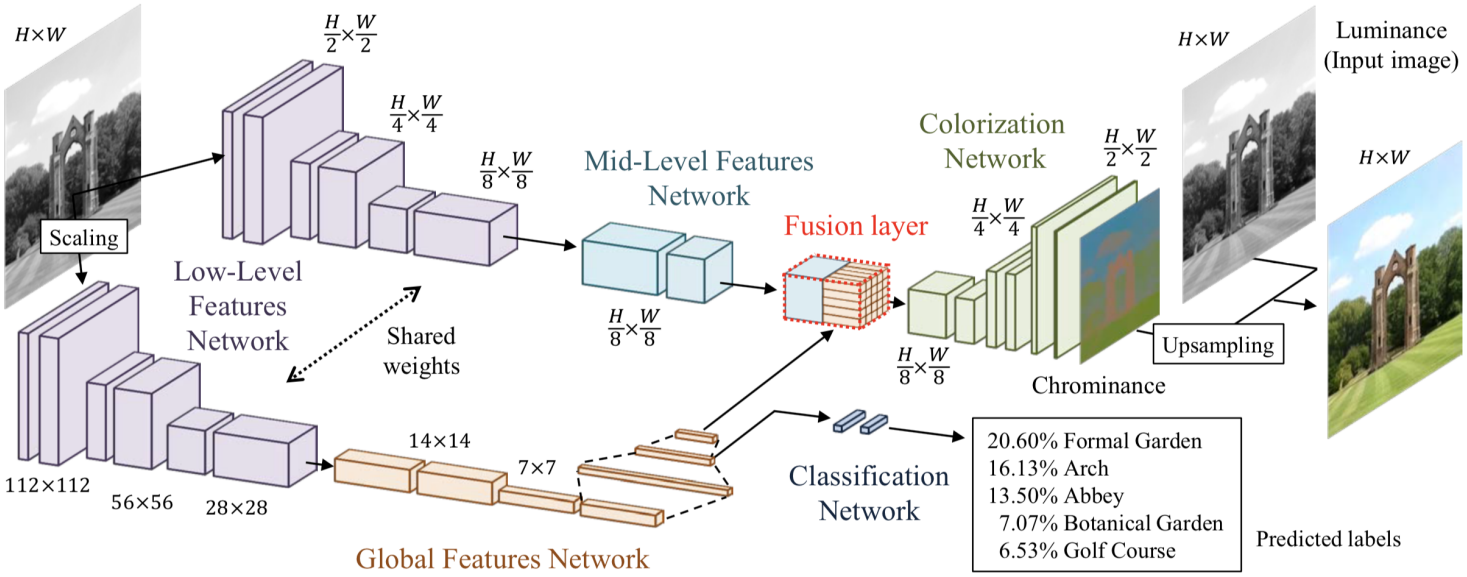
Image Colorization

- *CIE Lab* color-space, or just *Lab*
- The Lab color space describes mathematically all perceivable colors in the three dimensions
- L for lightness and a and b for the color components green–red and blue–yellow
- Lab color is designed to approximate human vision
- Using L component to adjust the lightness contrast
- Using a and b components to color balance corrections
- Conversions? $RGB \rightarrow XYZ \rightarrow Lab$, check: `http://www.brucelindbloom.com/index.html?Math.html`

Image Colorization

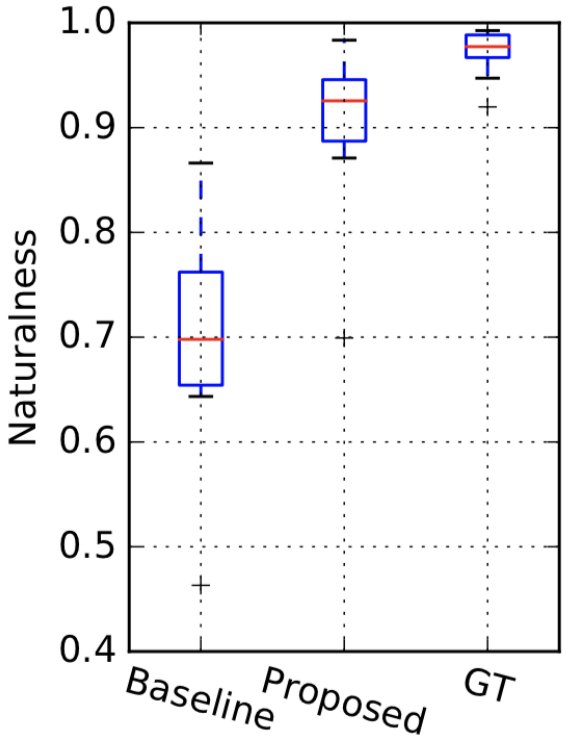
- Most methods fuse features from:
  - different representation levels
  - different networks
- Use a multi target loss

Image Colorization



$$L(y^{\text{color}}, y^{\text{class}}) = \|y^{\text{color}} - y^{\text{color},*}\|^2_{\text{FRO}} - \alpha \left(y^{\text{class}}_{l^{\text{class}}} - \log \left(\sum_{i=0}^{N} \exp \left(y^{\text{class}}_i \right) \right) \right)$$

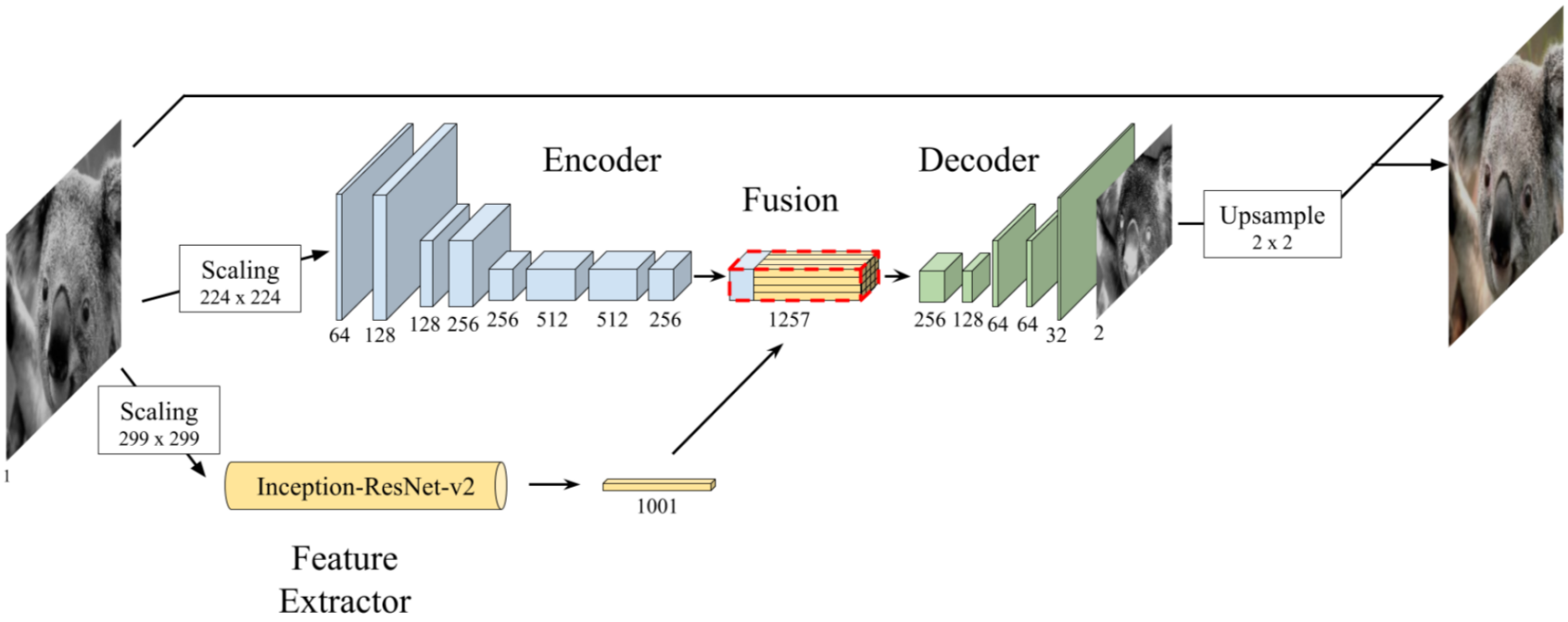
Image Colorization



Approach	Naturalness (median)
Ground Truth	97.7%
Proposed	92.6%
Baseline	69.8%

Image Colorization

- Simple loss from pre-trained model



$$C(\mathbf{X}, \boldsymbol{\theta}) = \frac{1}{2HW} \sum_{k \in \{a,b\}} \sum_{i=1}^{H} \sum_{j=1}^{W} (X_{k_{i,j}} - \tilde{X}_{k_{i,j}})^2$$

Image Colorization

- Fusion:

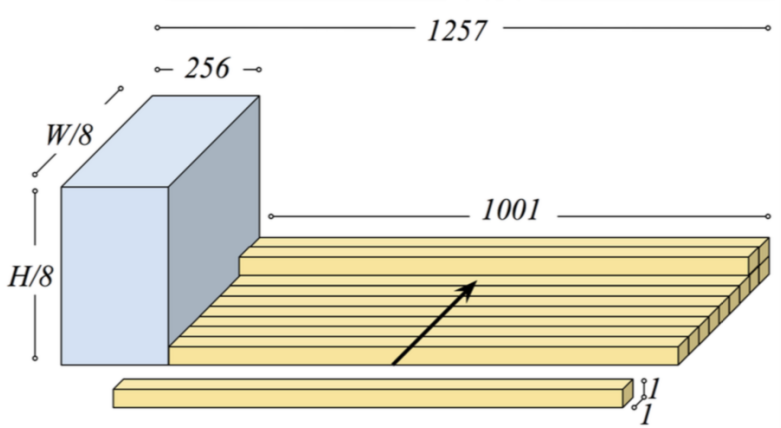
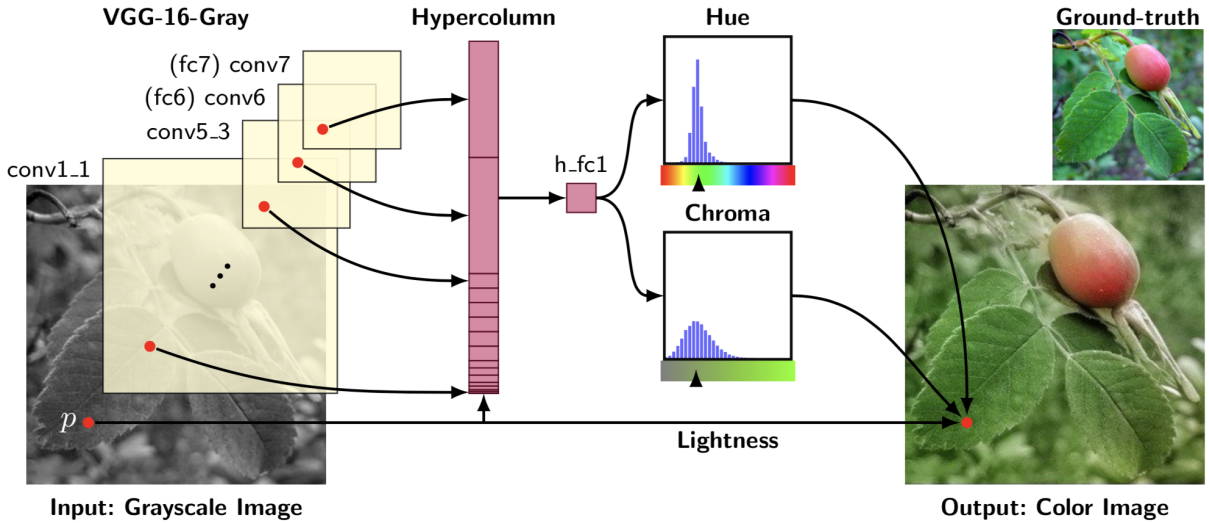


Image Colorization

- Using pre-trained VGG-16 and Hyper-columns representation



- Hue and Chroma loss for histograms of K bins *at every pixel*:

$$C(\mathbf{X}, \boldsymbol{\theta}) = \frac{1}{2HW} \sum_{k \in \{a,b\}} \sum_{i=1}^{H} \sum_{j=1}^{W} (X_{k_{i,j}} - \tilde{X}_{k_{i,j}})^2$$

- Taking expectation on inference.

Image Colorization

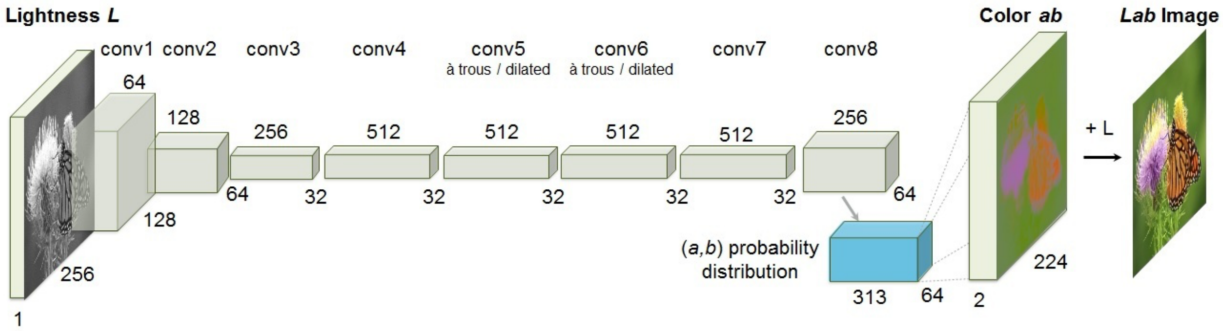
- Post-process using chromatic fading

Original

Chromatic fading

Image Colorization

- No pre-trained model



- Q bins inference
- Loss with weighted values for *rare* colors

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

Image Colorization

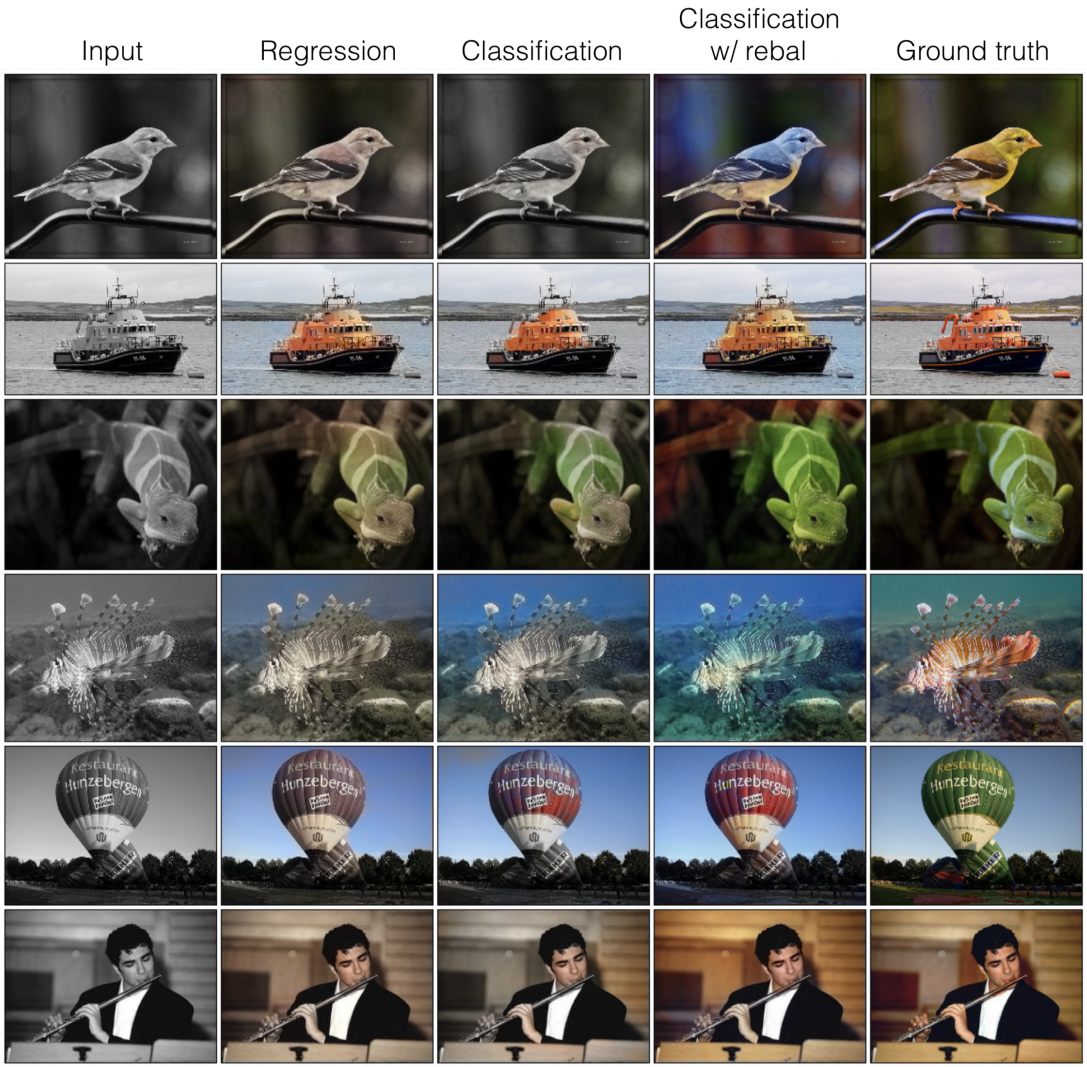


Image Colorization

Colorization Results on ImageNet							
Method	**Model Params (MB)**	**Feats (MB)**	**Runtime (ms)**	**AuC non-rebal (%)**	**AuC rebal (%)**	**VGG Top-1 Class Acc (%)**	**AMT Labeled Real (%)**
Ground Truth	–	–	–	100	100	68.3	50
Gray	–	–	–	89.1	58.0	52.7	–
Random	–	–	–	84.2	57.3	41.0	13.0±4.4
Dahl [2]	–	–	–	90.4	58.9	48.7	18.3±2.8
Larsson et al. [23]	588	495	122.1	**91.7**	65.9	**59.4**	**27.2±2.7**
Ours (L2)	129	127	17.8	91.2	64.4	54.9	21.2±2.5
Ours (L2, ft)	129	127	17.8	91.5	66.2	56.5	23.9±2.8
Ours (class)	129	142	22.1	91.6	65.1	56.6	25.2±2.7
Ours (full)	129	142	22.1	89.5	**67.3**	56.0	**32.3±2.2**

Image Colorization, Take Away

- Regression does not work. Multimodal problem.
- Treat as a classification problem (color space quantification)
- Potential merge different scales or learning levels
- Potential multi-part loss, classification and color space
- Some post-process could help

Image inpainting

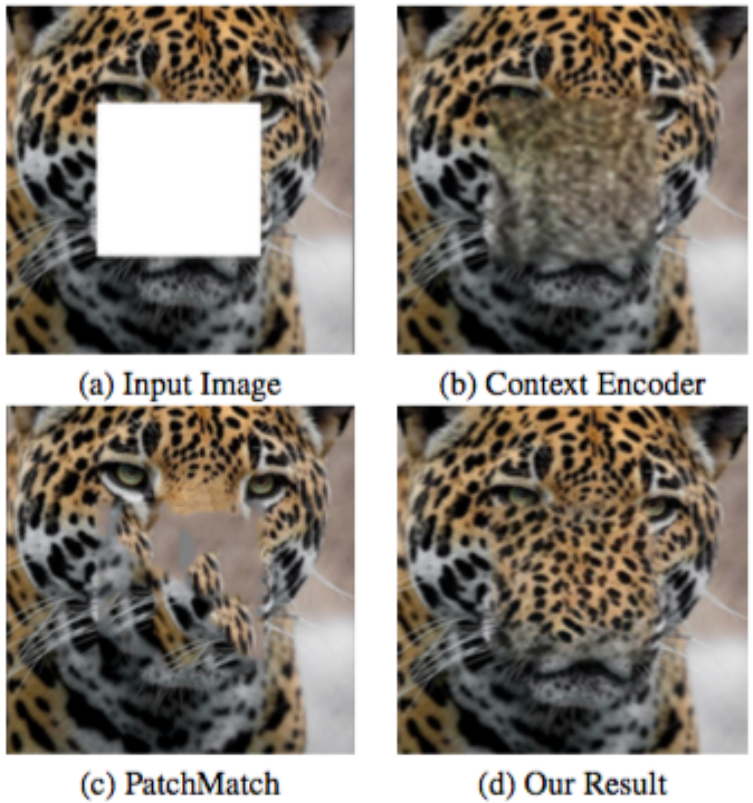
https://www.youtube.com/watch?v=psOPu3TldgY

Image inpainting

- Fill the gaps in visual information

Image inpainting

- Image inpainting must:
  - semantically plausible
  - context aware
  - without unpleasant boundaries
  - without blurry appearance



(a) Input Image (b) Context Encoder

(c) PatchMatch (d) Our Result

Image inpainting

- Existing methods:
  - Texture synthesis techniques. Extending textures from surrounding regions. Multi-scale, multi-orientation...
  - Data-driven, leveraging large external databases. Regions surrounded by similar context likely possess similar content.
- Modern techniques are based on DL like the Context Encoder.

Image inpainting, Context Encoder

- Context Encoders: Feature Learning by Inpainting
  - Train a CNN to regress the missing pixel values
  - Encoder captures the context of an image into a compact latent feature representation
  - Decoder uses that representation to produce the missing image content
  - Multi-part-loss: reconstruction loss and adversarial loss
  - Without adversarial loss we get blurry results

Image inpainting, Context Encoder

- Encoder captures the context of an image into a compact latent feature representation
- Decoder uses that representation to produce the missing image content
- Encoder comes from AlexNet, 5 first conv+pool to get $6 \times 6 \times 256$ feature map
- Not pre-trained
- Channel-wise Fully Connected layer between encoder and decoder, mn^4 instead of m^2n^4 for a m maps of $n \times n$ size.

Image inpainting, Context Encoder

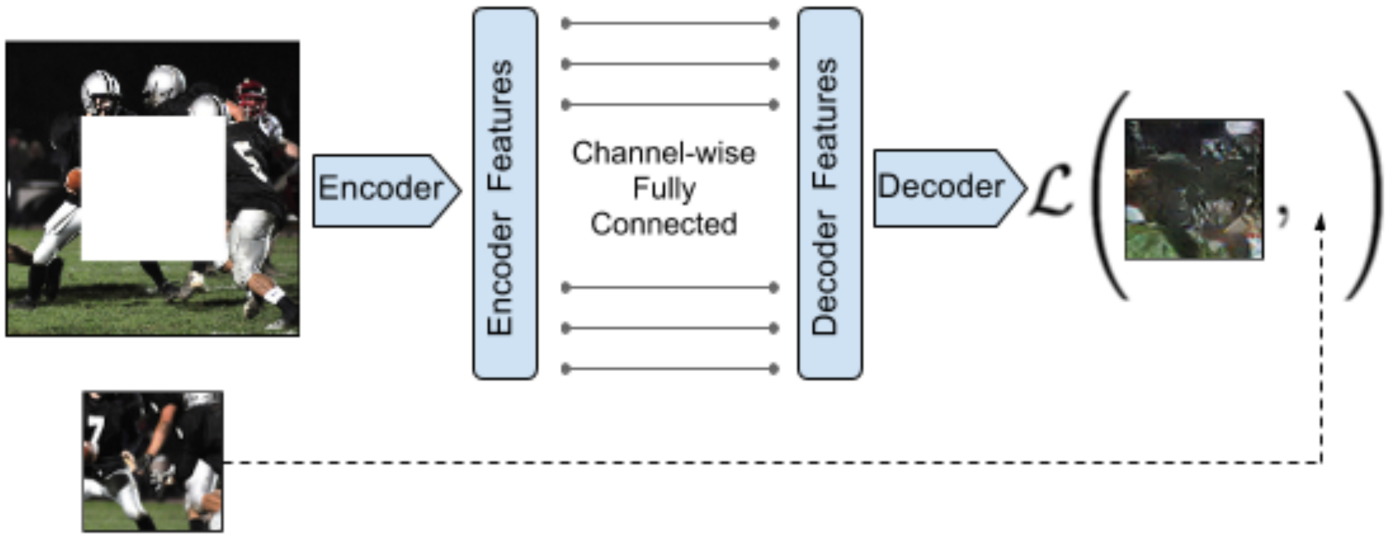


Image inpainting, Context Encoder

- Decoder generates pixels of the image using the encoder features
- After the Channel-wise Fully Connected layer five up-convolutional layers
- A up-convolutional is a convolution that results in a higher resolution image:
  - Up-sampling + Convolution
  - Convolution with fractional stride
- non-linear weighted up-sampling of the feature produced by the encoder

Image inpainting, Context Encoder

- Loss function, a joint loss function: reconstruction and adversarial
- The reconstruction loss is responsible for capturing the overall structure of the missing region and coherence with regards to its context
- The reconstruction loss tends to **average** together the multiple modes
- The adversarial loss, on the other hand, tries to make prediction look real, and has the effect of picking a particular mode from the distribution

Image inpainting, Context Encoder

- Reconstruction Loss
- Is a **masked** L_2 distance

$$L_{rec}(x) = || \, M \odot (x - F((1 - M) \odot x)) \, ||_2^2$$

- where $F(\cdot)$ is the result of the decoder
- M is a mask with 1 in the hidden regions
- and $\odot$ is the element wise product

Image inpainting, Context Encoder

- Adversarial Loss
- based on GAN with alternative formulation conditioning only the Generator

$$L_{adv}(x) = max_D \quad E_{x \in X}[log(D(x)) + log(1 - D(F((1 - M) \odot x)))]$$

- Finally a joint loss:

$$L = \lambda_{res} L_{rec} + \lambda_{adv} L_{adv}$$

Image inpainting, Context Encoder

- Some results

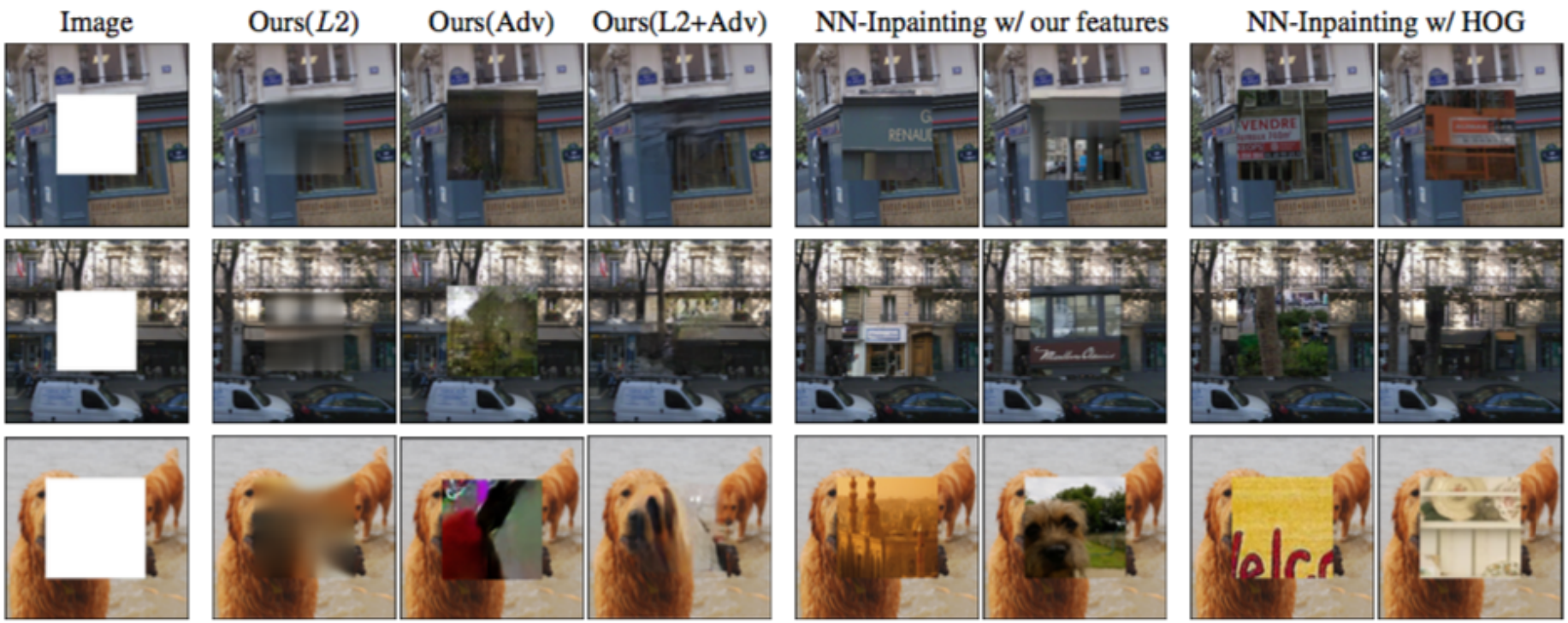


Image inpainting, A multi-scale approach

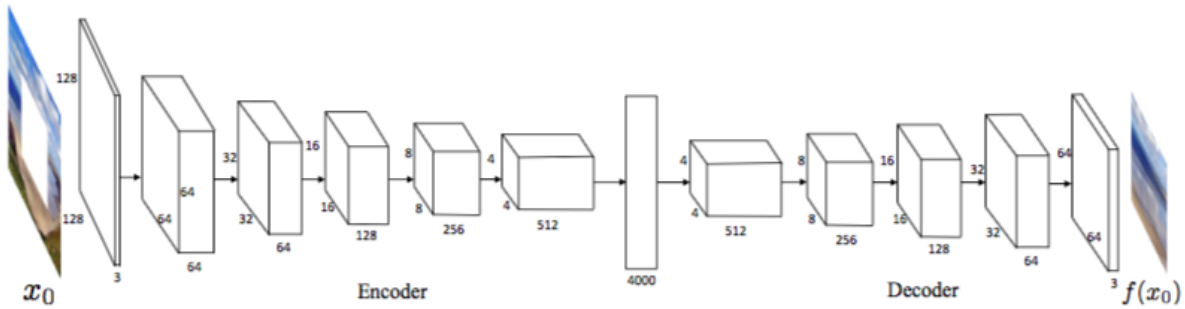
- High-Resolution Image Inpainting using Multi-Scale Neural Patch Synthesis:
  - Three level scale pyramid, e.g. 512,256,128
  - From coarse to fine, the result of one scale is used as input to the next
  - Scale $i+1$ consider output form scale i and results on $\tilde{x}_{i+1}$ following minimization problem:

$$\tilde{x}_{i+1} = \arg\min_{x} E_c(h(x,R), h(x_i,R)) + \alpha E_t(\phi_t(x), R^{\phi}) + \beta \Upsilon(x)$$

  - Two different networks: Content network and Texture network

Image inpainting, A multi-scale approach

- Content network



- The first term E_c models the holistic content constraint

$$E_c(h(x,R),h(x_i,R)) = || \; h(x,R) - h(x_i,R) \; ||$$

- $h(x,R)$ is the output of the network and $h(x_i,R)$ the prediction in the coarser scale
- For the initial scale they use a Context Network
- E_c penalize the l_2 difference between the optimization result and the previous content prediction

Image inpainting, A multi-scale approach

- Texture network
- Use a VGG-19 to obtain any map $h(\phi_t(x), P_i)$ for the second term $E_t(\phi_t(x), R^\phi)$
- penalizes the discrepancy of the texture appearance inside and outside the hole

$$E_t(\phi_t(x), R^\phi) = \frac{1}{\mid R^\phi \mid} \sum_{i \in R^\phi} \mid\mid h(\phi_t(x), P_i) - h(\phi_t(x), P_{nn(i)}) \mid\mid_2^2$$

- P are potential patches inside the region R in the feature map ϕ_t
- $P_{nn(i)}$ is the most similar patch of neighboring locations outside the region R

Image inpainting, A multi-scale approach

- Results

Image super-resolution

- Recover a high-resolution image from a single low-resolution image
- A classical computer vision problem
- This problem is inherently ill-posed since a multiplicity of solutions exist for any given low-resolution pixel
- it is an under-determined inverse problem, of which solution is not unique

Image super-resolution

https://www.youtube.com/watch?v=nKtE-V6LNpE

Image super-resolution. Deep Learning problems

- Consider this image:

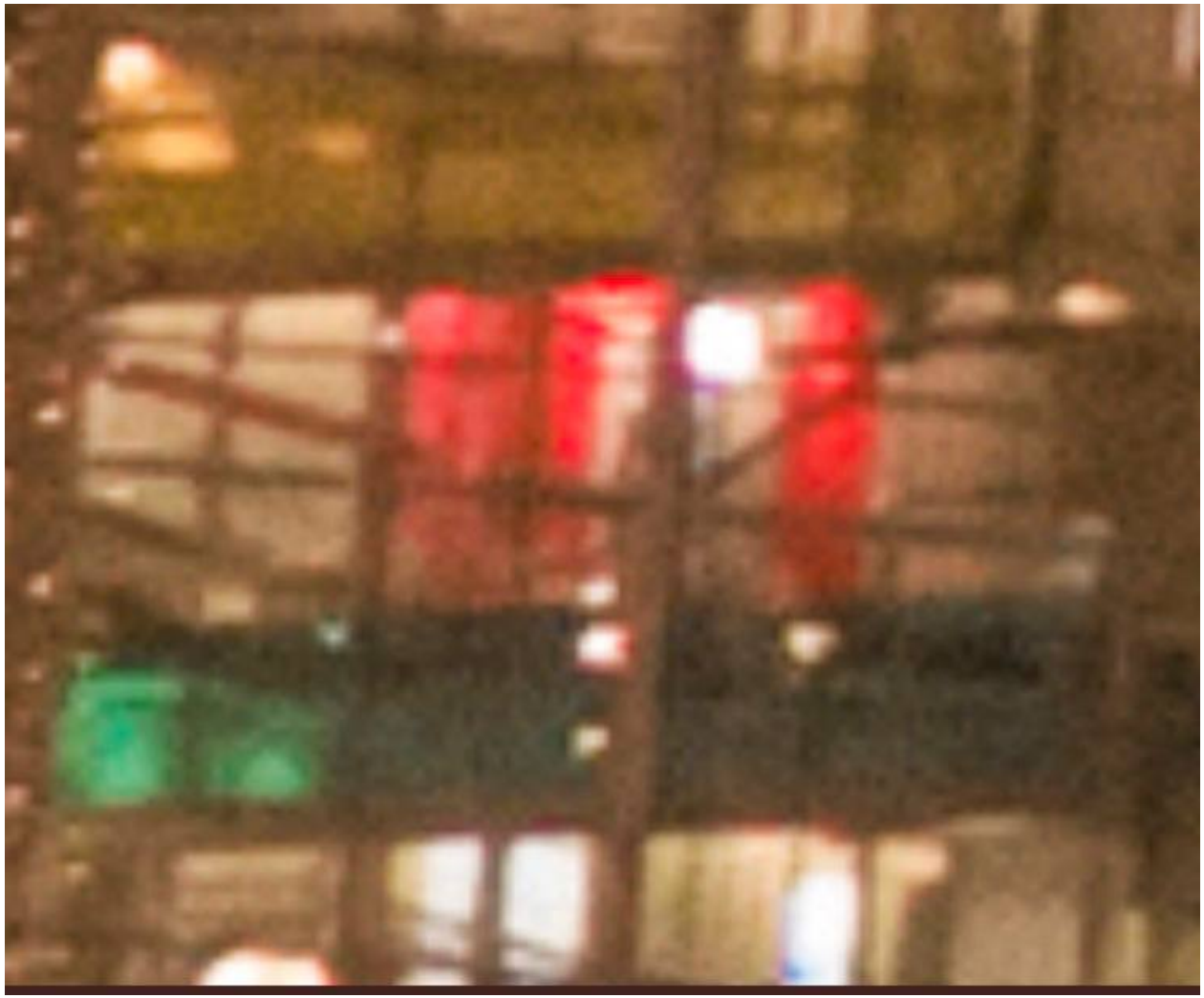


Image super-resolution. Deep Learning problems

- See the result:

Image super-resolution

- The Super-Resolution Convolutional Neural Network (SRCNN)
- End-to-end mapping between low- and high-resolution images
- Relationships with classical sparse-coding-based methods
- Use all color channels
- Preprocessing: Up-scale to the desire size with bi-cubic interpolation
- Recover the high-resolution image form this interpolation
- Training to minimize MSE

Image super-resolution

- Results

Eval. Mat	Scale	Bicubic	SC [50]	NE+LLE [4]	KK [25]	ANR [41]	A+ [41]	SRCNN
PSNR	2	33.66	-	35.77	36.20	35.83	36.54	**36.66**
	3	30.39	31.42	31.84	32.28	31.92	32.59	**32.75**
	4	28.42	-	29.61	30.03	29.69	30.28	**30.49**
SSIM	2	0.9299	-	0.9490	0.9511	0.9499	**0.9544**	0.9542
	3	0.8682	0.8821	0.8956	0.9033	0.8968	0.9088	**0.9090**
	4	0.8104	-	0.8402	0.8541	0.8419	0.8603	**0.8628**
IFC	2	6.10	-	7.84	6.87	8.09	**8.48**	8.05
	3	3.52	3.16	4.40	4.14	4.52	**4.84**	4.58
	4	2.35	-	2.94	2.81	3.02	**3.26**	3.01
NQM	2	36.73	-	42.90	39.49	43.28	**44.58**	41.13
	3	27.54	27.29	32.77	32.10	33.10	**34.48**	33.21
	4	21.42	-	25.56	24.99	25.72	**26.97**	25.96
WPSNR	2	50.06	-	58.45	57.15	58.61	**60.06**	59.49
	3	41.65	43.64	45.81	46.22	46.02	**47.17**	47.10
	4	37.21	-	39.85	40.40	40.01	41.03	**41.13**
MSSSIM	2	0.9915	-	0.9953	0.9953	0.9954	**0.9960**	0.9959
	3	0.9754	0.9797	0.9841	0.9853	0.9844	**0.9867**	0.9866
	4	0.9516	-	0.9666	0.9695	0.9672	0.9720	**0.9725**

Image super-resolution

- Image Restoration Using Very Deep Convolutional Encoder-Decoder Networks with Symmetric Skip Connections
- Convolutional and Deconvolutional (ConvT) network
- Deep residual encoder decoder. Deep RED.
- No pooling (or Unpooling)
- 64@3 $\times$ 3 kernels (inspired in VGG) and skip connections (Residual)

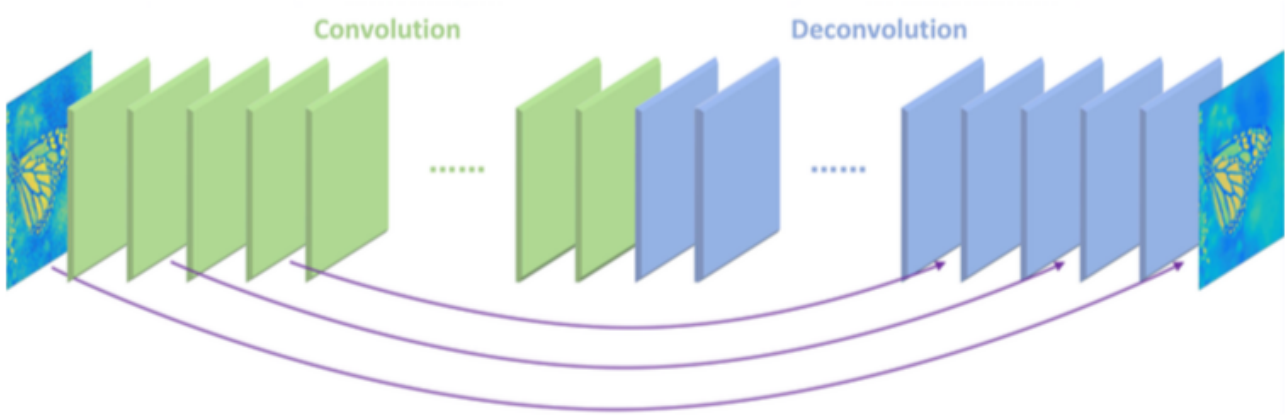


Image super-resolution

- Results

	PSNR								
	SRCNN	NBSRF	CSCN	CSC	TSE	ARFL+	RED10	RED20	RED30
$s = 2$	36.66	36.76	37.14	36.62	36.50	36.89	37.43	37.62	**37.66**
$s = 3$	32.75	32.75	33.26	32.66	32.62	32.72	33.43	33.80	**33.82**
$s = 4$	30.49	30.44	31.04	30.36	30.33	30.35	31.12	31.40	**31.51**
	SSIM								
$s = 2$	0.9542	0.9552	0.9567	0.9549	0.9537	0.9559	0.9590	0.9597	**0.9599**
$s = 3$	0.9090	0.9104	0.9167	0.9098	0.9094	0.9094	0.9197	0.9229	**0.9230**
$s = 4$	0.8628	0.8632	0.8775	0.8607	0.8623	0.8583	0.8794	0.8847	**0.8869**

Image super-resolution

- VDSR

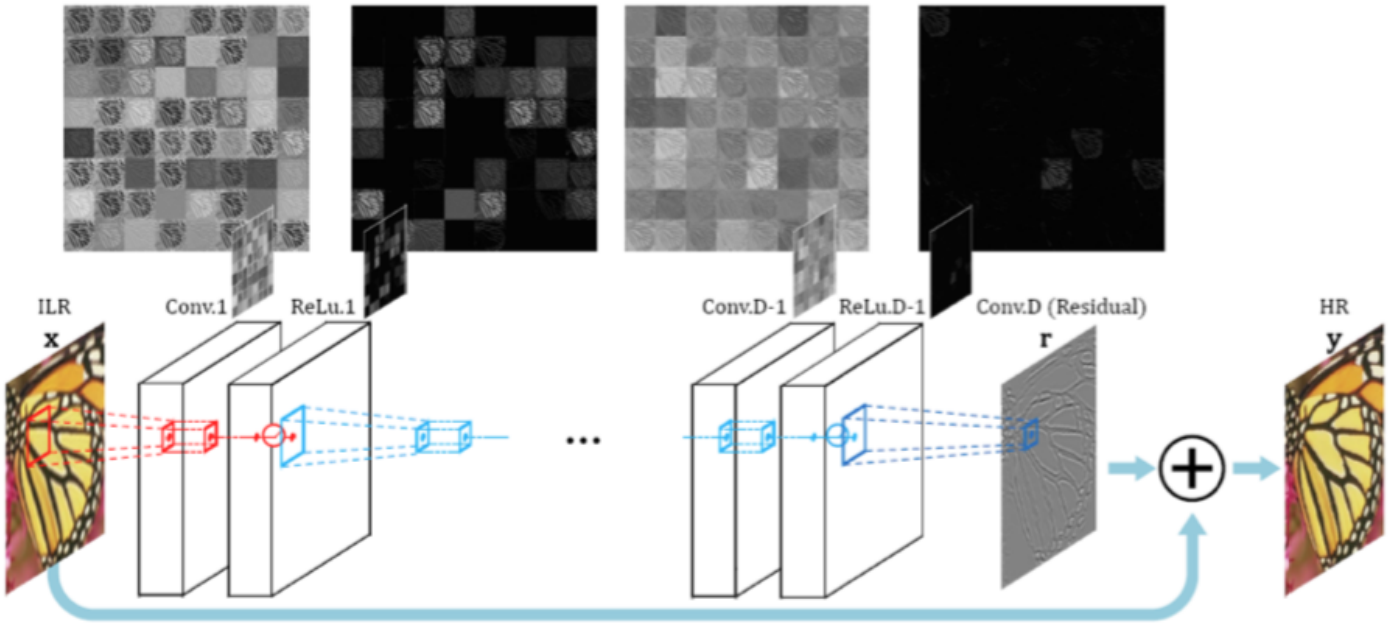


Image super-resolution

- MSCN

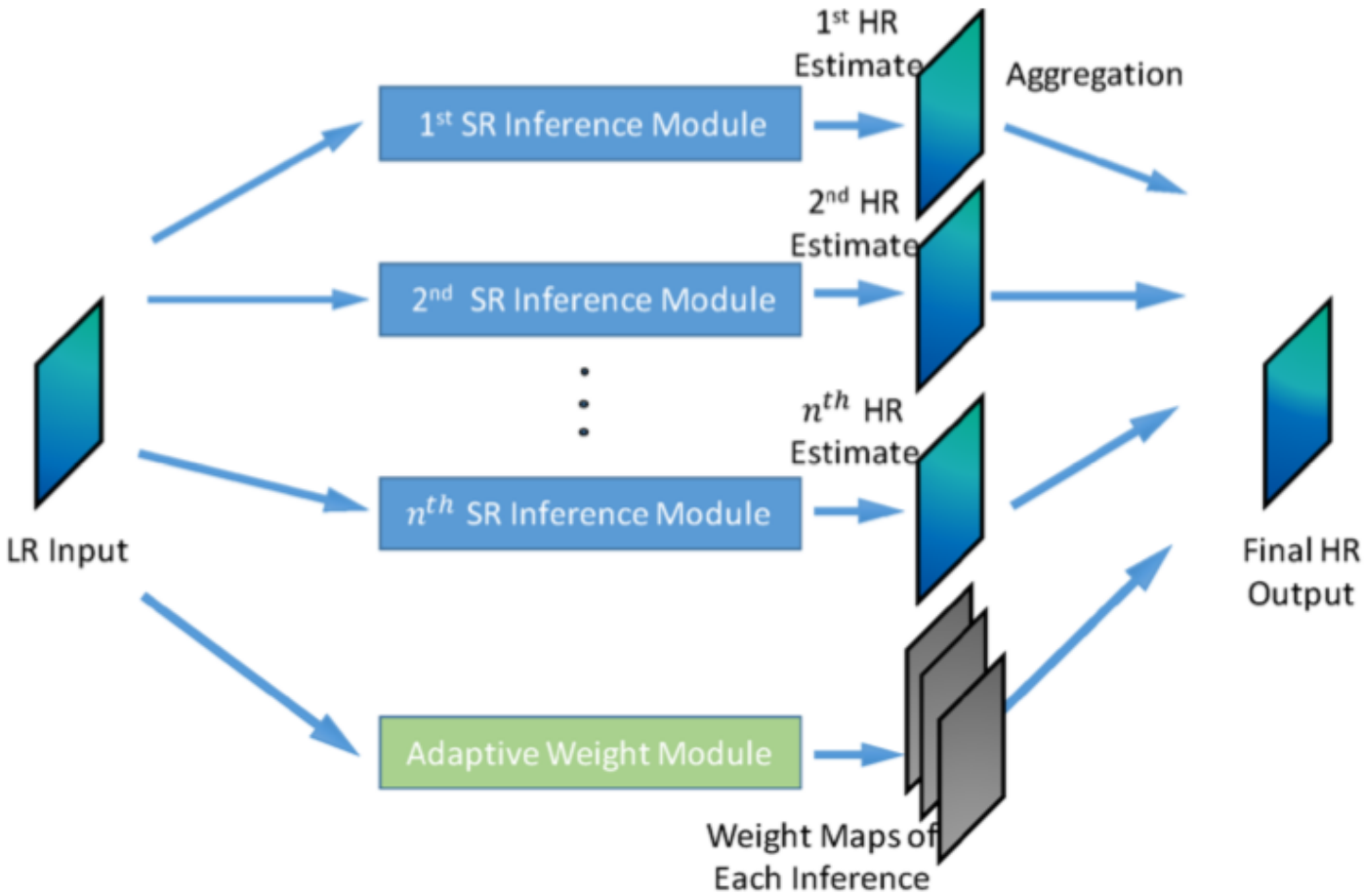


Image super-resolution

- LSP/HSP

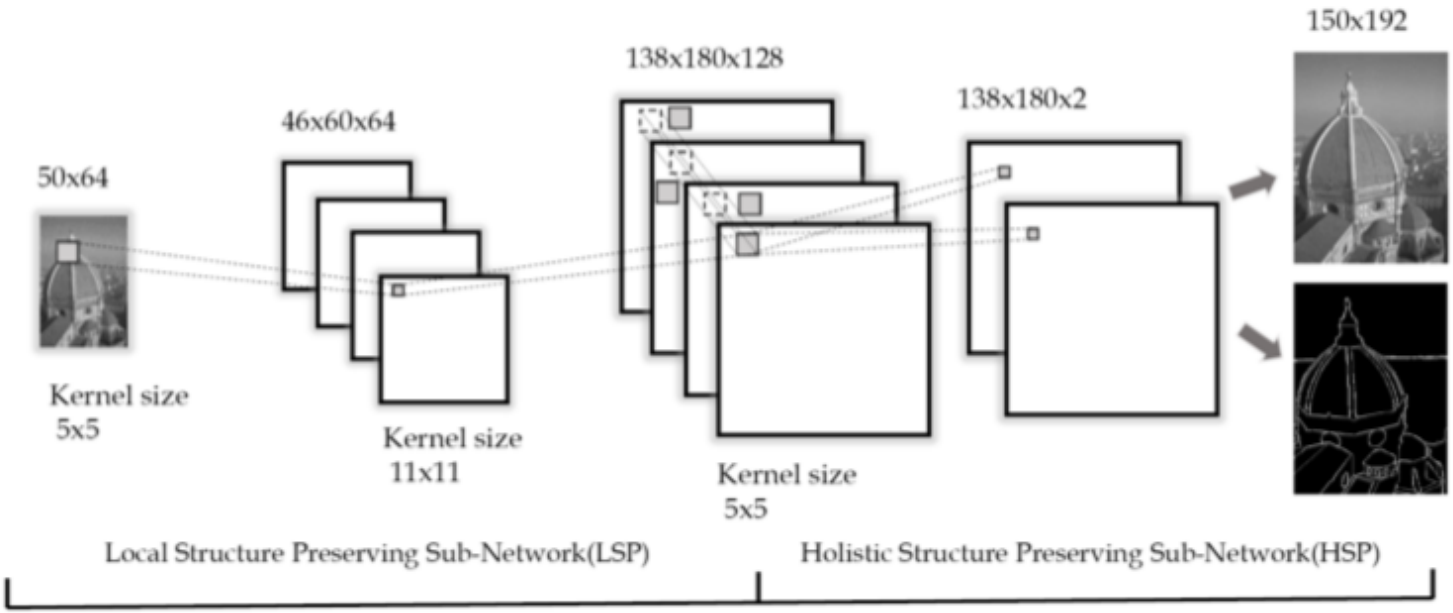


Image super-resolution

- GUN

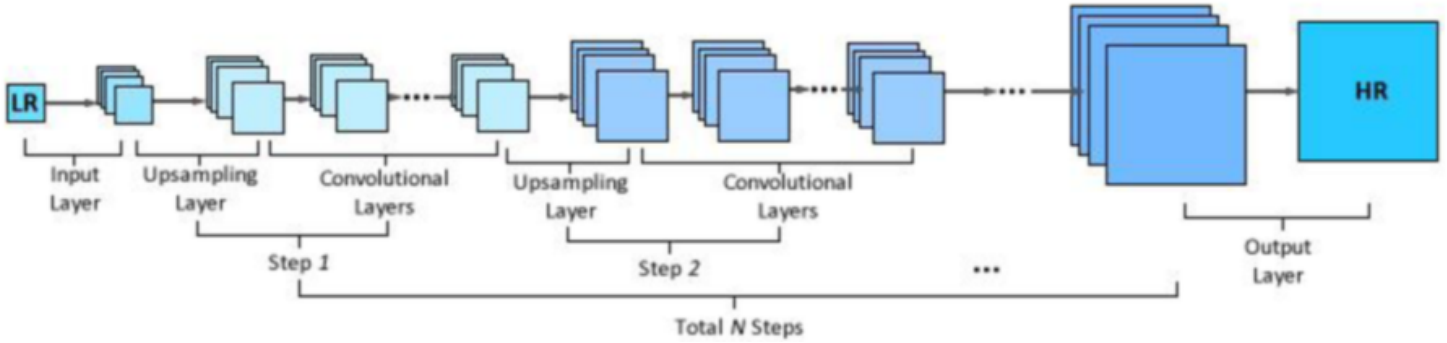


Image super-resolution

- EEDS

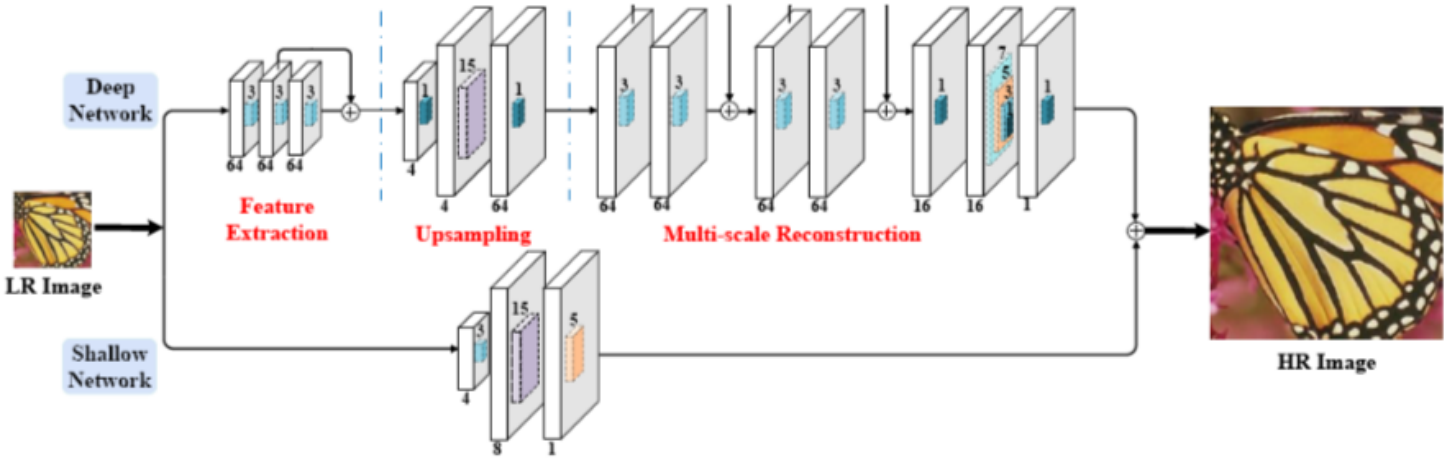


Image super-resolution

- Results

S/No.	Name with reference	PSNR (dB)
1	Bicubic	27.54
2	SC [39]	28.31
3	K-SVD [91]	28.67
4	ANR [46]	28.65
5	A+ [47]	29.13
6	NE+LLE [45]	28.60
7	KK [86]	28.94
8	SRCNN-Ex [4]	29.00
9	SRCNN [5]	29.30
10	FSRCNN [69]	29.43
11	DPN [104]	29.80
12	ShCNN [123]	29.39
13	Self-Ex [70]	29.16
14	VDSR [51]	29.77
15	DRCN [63]	29.76
16	HWCN [132]	29.17
17	SCN [100]	29.41
18	CSCN [102]	29.55
19	MSCN-4 [103]	29.65
20	RED30 [110]	29.61
21	DSRCNN [111]	28.60
22	NBSRF [49]	29.25
23	R-basic [52]	29.67
24	R-deep [52]	29.80
25	RFL [48]	29.05
26	ARFL [48]	29.13
27	RFL+ [48]	29.17
28	ARFL+ [48]	29.23
29	ESPCN [118]	29.49
30	IA [67]	29.69
31	DJSR [108]	29.96