



河南大學

明德新民 止于至善

智能图像修复与图像彩化处理

陈小潘

计算机与信息工程学院

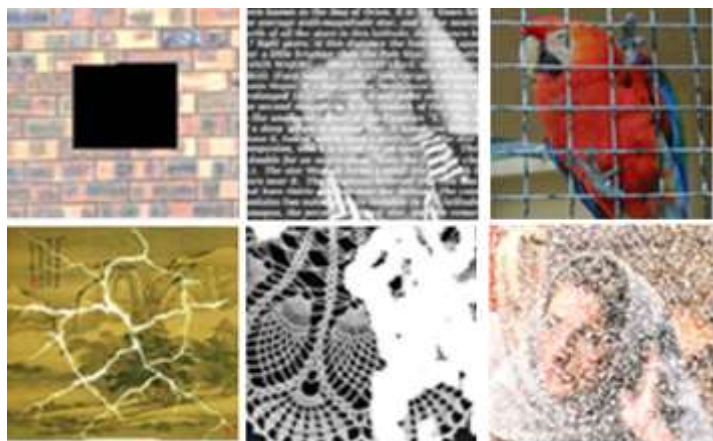


提纲

- 智能图像修复
- 智能彩色图像处理

图像修复相关概念

- 图像的重建与修复是指对于低质图像的恢复过程。
- 图像修复一般指对图像丢失或损坏的部分进行修补，使图像恢复原貌。
- 图像重建具有更一般的复原含义，对图像破损、受到信号干扰的降质，以及由于采集设备低质量引起的劣质图像的重构都属于图像的重建。
- 常见的图像损坏包括目标遮挡、文本遮挡、噪声、照片划痕、图像采集带来的高光效应等。



图像修复相关概念

- 对于图像修复，为了对图像中破损部分进行修补，在传统技术中出现了图像补绘（Image Inpainting）和空洞填充（Region Filling）等技术。
- 传统修复技术研究中是设法利用替换或者填充手段，按照不同的规则，将周围非破损区域的像素信息填充或者编辑到损坏区，以便达到修补后的外观一致性。
- 近年来，随着人工智能技术的迅猛发展，出现了基于深度学习的图像修复技术，包括基于深度学习的Inpainting技术以及基于深度学习的超分辨率复原技术。

1.智能图像的修复技术发展

基于深度学习的图像补绘技术

- 2016年，Deepak Pathak等提出基于上下文信息感知的图像补绘方法（Context Encoders: Feature Learning by Inpainting），采用无监督学习。
- 2017年Chao Yang等提出了基于结合图像内容和纹理进行约束优化的多尺度补绘方法，不仅仅考虑上下文的感知信息，而且根据中间层特征的相似性匹配，取得满意的结果。
- 2018年，Guilin Liu等提出有效算法Image Inpainting for Irregular Holes Using Partial Convolutions，通过部分卷积和掩膜更新代替卷积层，可以修补任意形状的破损区域。

1.智能图像的修复技术发展

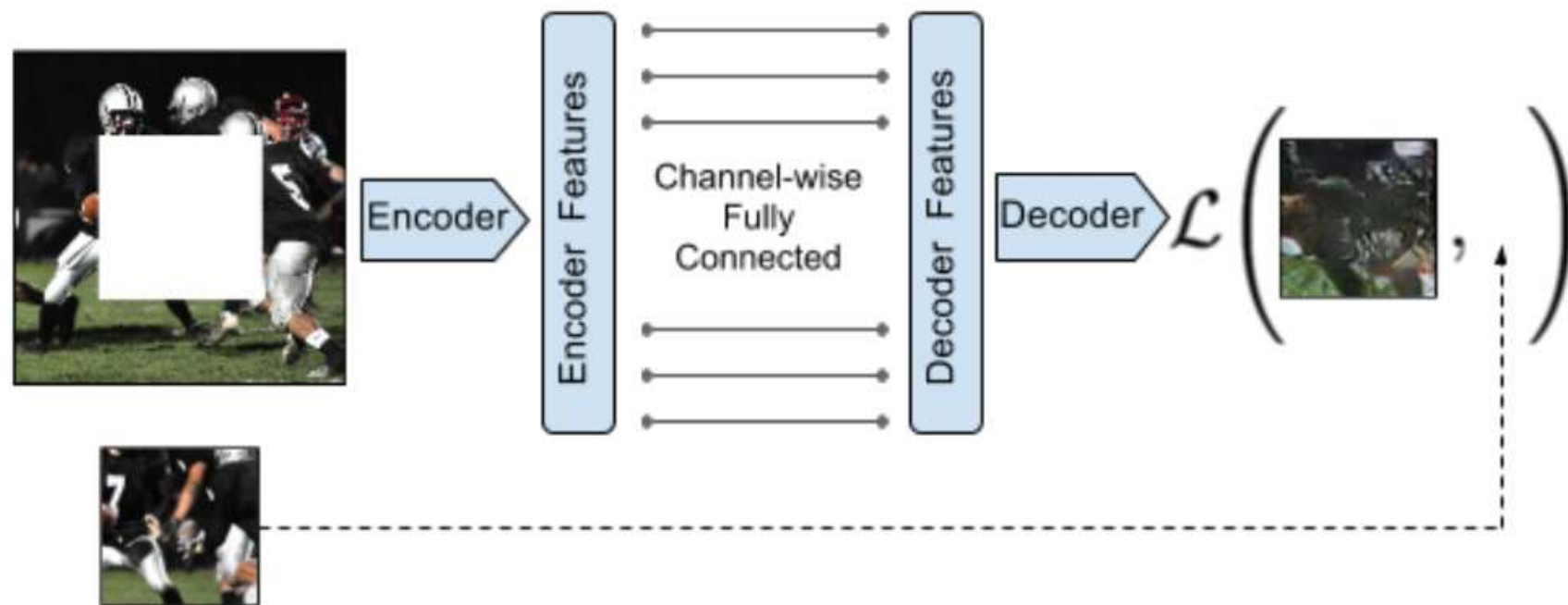
基于深度学习的图像补绘技术

- 2019年，Kamyar Nazeri提出一种基于两阶段（two-stage）生成对抗网络的边缘补全图像修复方法，借助启发式生成模型得到了缺失部分边缘信息，再将其作为先验，实现图像修复。
- 2020年，Ziyu Wan等提出一种模糊、褶皱照片的修复算法，用变分自动编码器，将图像变换到隐藏空间，在隐空间进行图像修复处理。
- 2021，Syed Waqas Zamir等人提出基于注意力机制的多阶段图像重建算法，采用编码解码的网络结构，对不同层级特征融合，实现高分辨率的重建。

2.基于上下文编码的算法

- Deepak Pathak等于2016年在CVPR国际会议上提出一种基于上下文信息感知的图像补绘方法Context Encoders: Feature Learning by Inpainting。
 - ✓ 采用有监督学习，设计编码和解码结构的CNN，对周围像素的上下文信息进行特征学习，进一步推断缺失像素的色彩信息。
 - ✓ 在判别器设计时，利用真值标签图像作为判别的依据，当网络生成结果与真值标签图像一致时，停止对网络参数的训练。

2. 基于上下文编码的算法



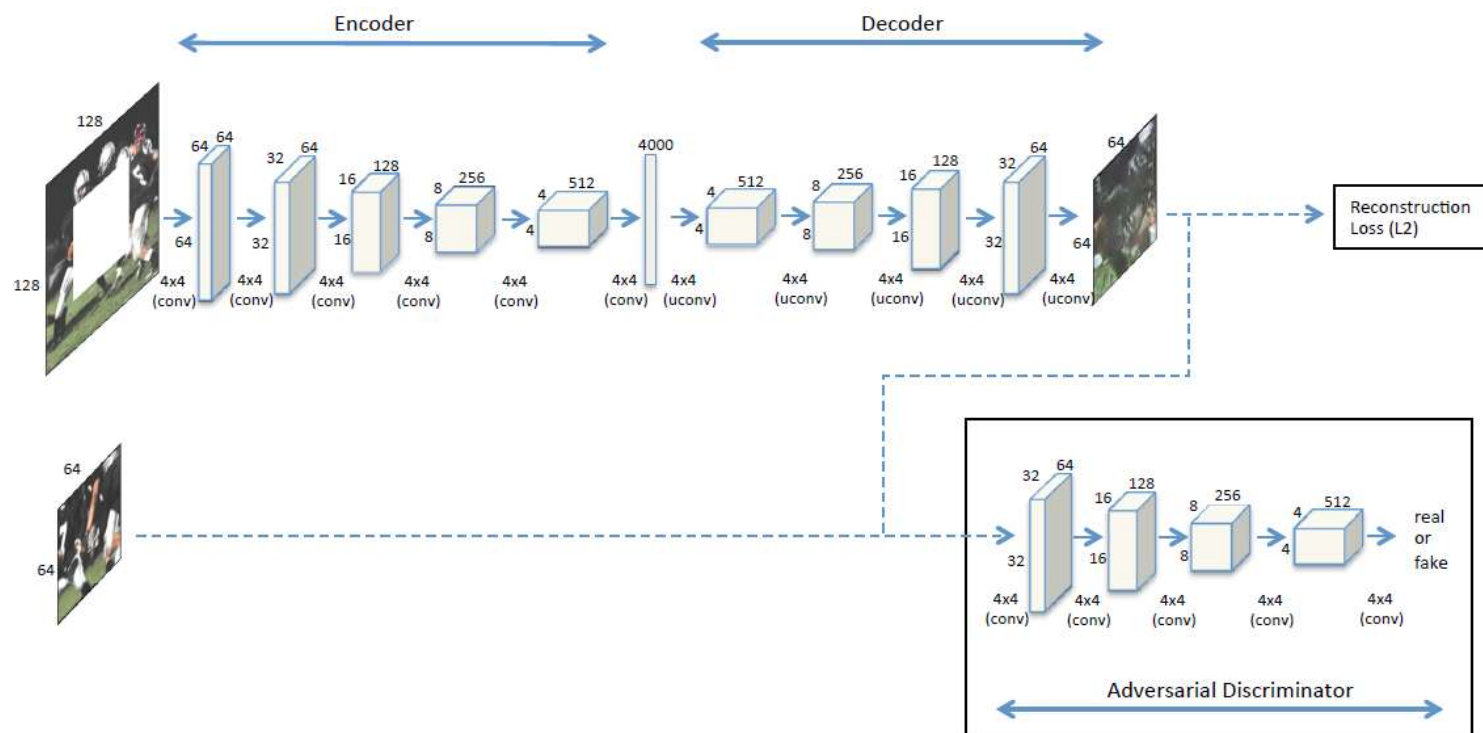
2.基于上下文编码的算法

创新点主要有两个：

- 使用了Encoder-decoder结构来实现图像修复，并弃用了全连接层，采用channel-wise的方式，减少了模型的参数；
- 损失函数包括两部分，一部分图像的重建损失，另一部分是GAN的对抗损失。

2.基于上下文编码的算法

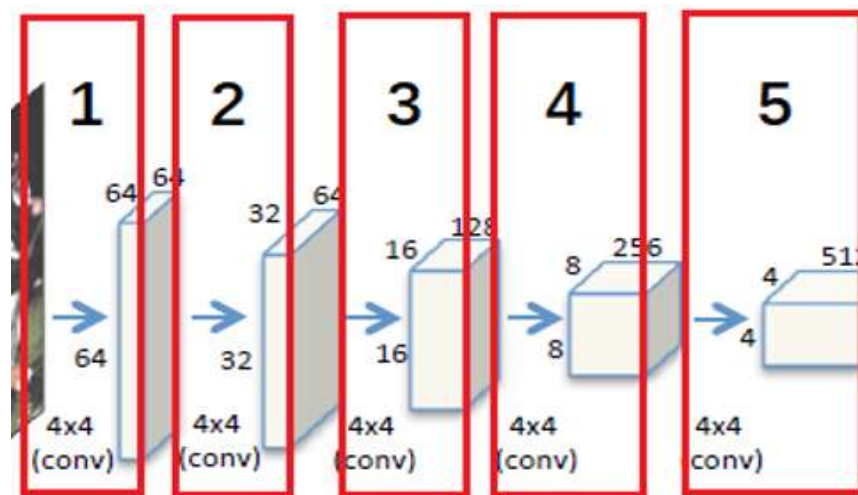
网络设计时，结合CNN和GAN的结构：去掉了ALexNet的全连接层，设计为卷积层。去掉所有现有ALexNet中的pooling层，而采用转置卷积进行上采样。



(a) Context encoder trained with joint reconstruction and adversarial loss for semantic inpainting. This illustration is shown for *center region dropout*. Similar architecture holds for arbitrary region dropout as well. See Section 3.2. https://blog.csdn.net/walxin_47444348

2.基于上下文编码的算法

编码器设计为6层（第1-6层）的卷积编码结构，第1至第5层由卷积运算处理，如图所示。卷积核的形状分别为 $4 \times 4 \times 64$ 、 $4 \times 4 \times 64$ 、 $4 \times 4 \times 128$ 、 $4 \times 4 \times 256$ 、 $4 \times 4 \times 512$



第1至第5层由卷积运算处理

2.基于上下文编码的算法

解码器设计为5层，如图所示

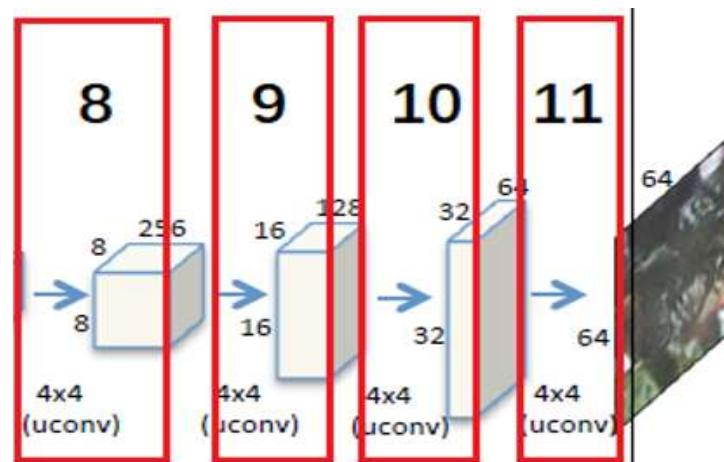


图 解码器设计

2.基于上下文编码的算法

- **重建损失**：采用输出图像与目标图像对应填补区域的L2损失作为缺失区域的损失。

$$\mathcal{L}_{rec}(x) = \| \hat{M} \odot \left(x - F \left((1 - \hat{M}) \odot x \right) \right) \|_2^2$$

其中 \hat{M} 表示被遮挡区域的掩膜， F 表示Context Encoder网络。 \odot 是矩阵对应位置相乘

- **判别损失**：采用交叉熵作为判别损失，使预测结果更加真实。

$$\mathcal{L}_{adv} = \max_D \mathbb{E}_{x \in \mathcal{X}} \left[\log(D(x)) + \log \left(1 - D \left(F \left((1 - \hat{M}) \odot x \right) \right) \right) \right]$$

- **重建损失与判别损失相结合**：图像的修补结果既需要考虑修补结果的语义的正确性，同时也要做到修补区域与周围区域具有相关性。

$$\mathbf{L} = \lambda_{rec} \mathbf{L}_{rec} + \lambda_{adv} \mathbf{L}_{adv}$$

2. 基于上下文编码的算法

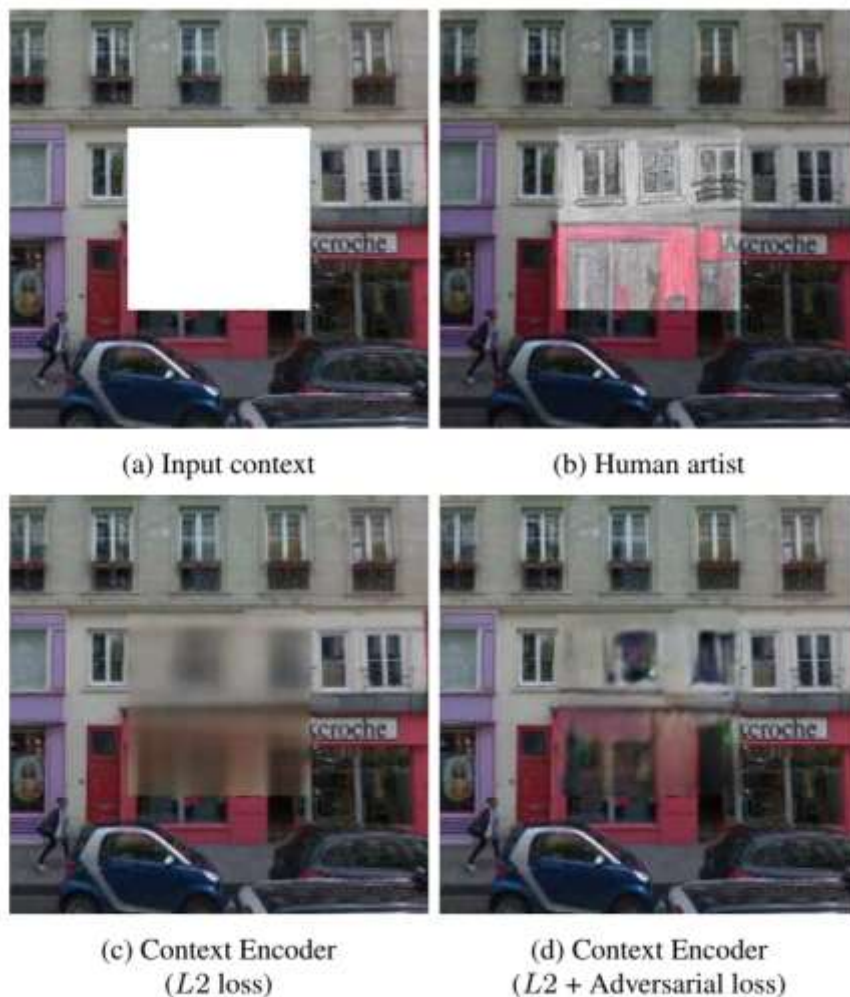
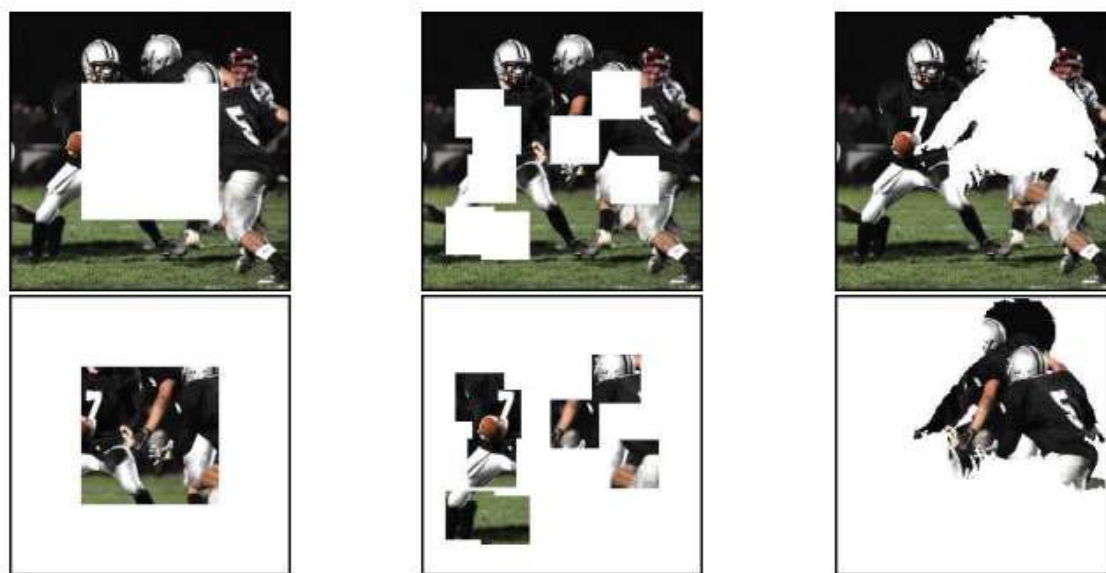


Figure 1: Qualitative illustration of the task. Given an image with a missing region (a), a human artist has no trouble inpainting it (b). Automatic inpainting using our *context encoder* trained with $L2$ reconstruction loss is shown in (c), and using both $L2$ and adversarial losses in (d).

2. 基于上下文编码的算法



(a) Central region

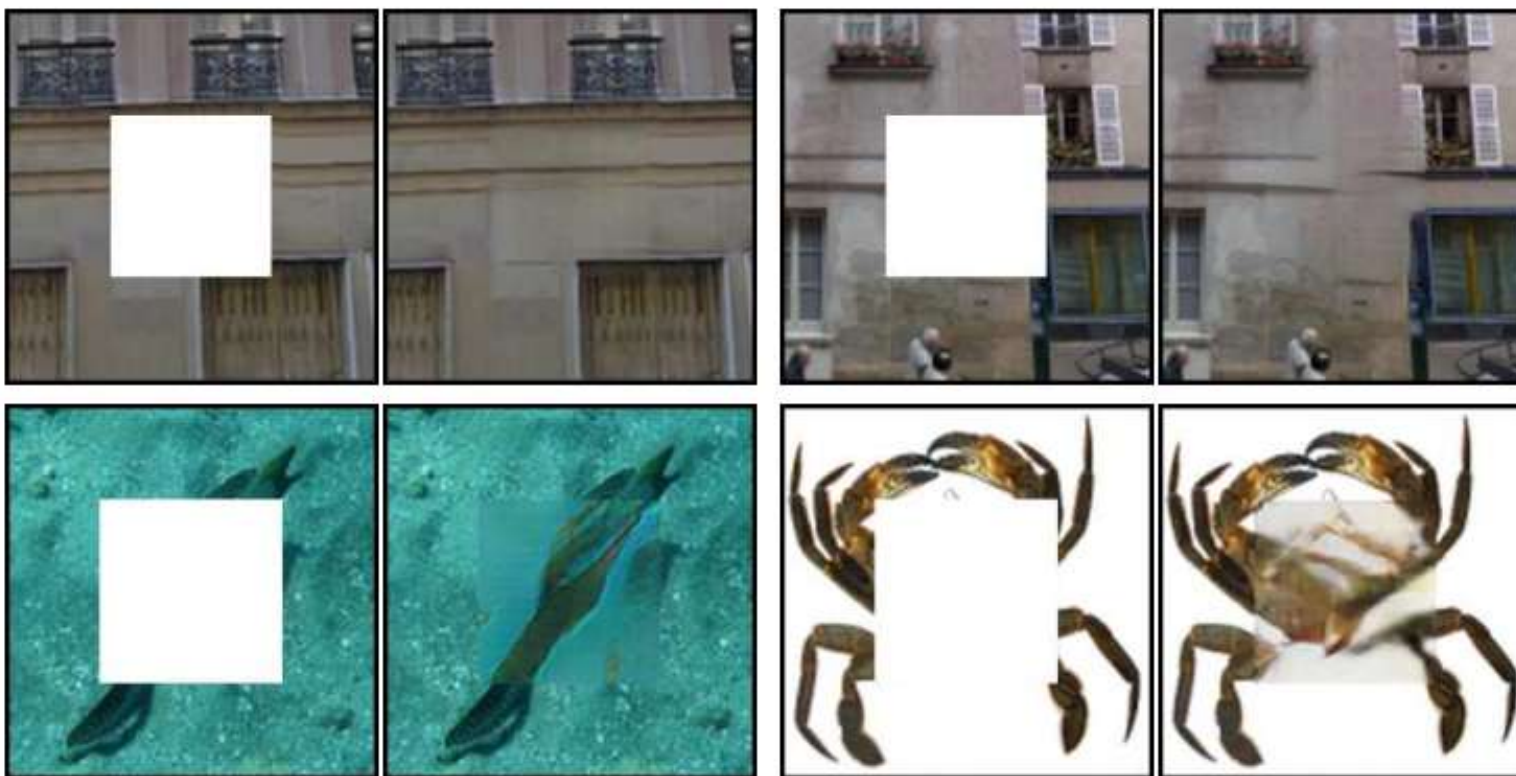
(b) Random block

(c) Random region

Figure 3: An example of image x with our different region masks \hat{M} applied, as described in Section 3.3.

2.基于上下文编码的算法

上下文信息编码的图像修补算法的实验结果



2.基于上下文编码的算法

从实验结果来看，与当时的其他方法相比还是很不错的，可以修复较大的缺失区域。但是可以发现局部的一些细节修复的不是很好，存在局部模糊的情况。

3.全局与局部一致性修补算法

- 2017年由早稻田大学(Waseda University)的学者Satoshi Iizuka（佐藤一裕）等人提出一种基于全局和局部一致性的图像修补算法（Globally and Locally Consistent Image Completion）。
- 利用GAN结构进行时在设计判别器时，采用了全局判别器（Global Discriminator, G-dis）和局部判别器（Local Discriminator, L-Dis）。全局鉴别器将完整图像作为输入，识别场景的全局一致性，而局部鉴别器仅在以填充区域为中心的原图像1/4大小区域上观测，识别局部一致性。
- 通过采用两个不同的鉴别器，使得最终网络不但可以使全局观测一致，并且能够优化细节，最终产生好的图片填充效果。两种判别器可以提高局部区域的清晰度和对比度。

3.全局与局部一致性修补算法

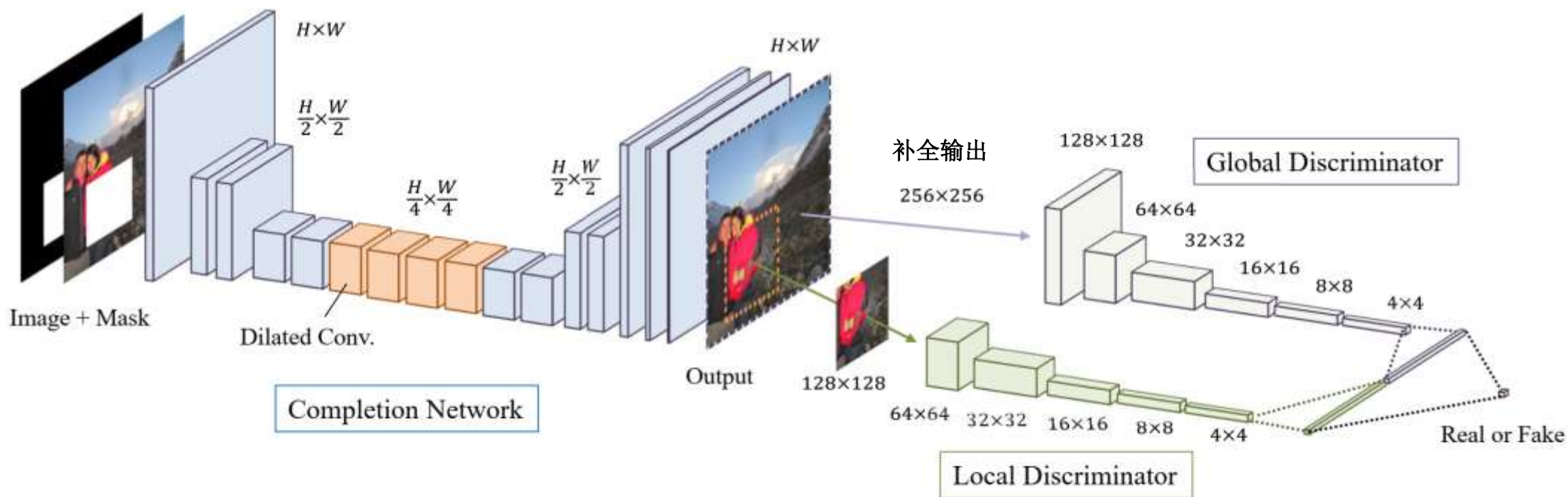


Fig. 2. Overview of our architecture for learning image completion. It consists of a completion network and two auxiliary context discriminator networks that are used only for training the completion network and are not used during the testing. The global discriminator network takes the entire image as input, while the local discriminator network takes only a small region around the completed area as input. Both discriminator networks are trained to determine if an image is real or completed by the completion network, while the completion network is trained to fool both discriminator networks.

3.全局与局部一致性修补算法

生成图像网络（completion network）没有使用全连接层，可以接受任意尺寸的输入图像，其采用12层卷积网络对原始图片(去除需要进行填充的部分)进行encoding，得到一张原图1/16大小的网格。然后再对该网格采用4层卷积网络进行decoding，从而得到复原图像，下表为生成网络各层参数分布情况。

- 输入：RGB图像，二进制掩码（需要填充的区域以1填充）
- 输出：RGB图像

Type	Kernel	Dilation (η)	Stride	Outputs
conv.	5×5	1	1×1	64
conv.	3×3	1	2×2	128
conv.	3×3	1	1×1	128
conv.	3×3	1	2×2	256
conv.	3×3	1	1×1	256
conv.	3×3	1	1×1	256
dilated conv.	3×3	2	1×1	256
dilated conv.	3×3	4	1×1	256
dilated conv.	3×3	8	1×1	256
dilated conv.	3×3	16	1×1	256
conv.	3×3	1	1×1	256
conv.	3×3	1	1×1	256
deconv.	4×4	1	$1/2 \times 1/2$	128
conv.	3×3	1	1×1	128
deconv.	4×4	1	$1/2 \times 1/2$	64
conv.	3×3	1	1×1	32
output	3×3	1	1×1	3

3.全局与局部一致性修补算法

全局鉴别网络输入是 256×256 ，RGB三通道图像，局部网络输入是 128×128 ，RGB三通道图像，根据论文当中的设置，全局网络和局部网络都会通过 5×5 的convolution layer，以及 2×2 的stride降低分辨率，最终分别得到1024维向量。然后，作者将全局和局部两个鉴别器输出连接成一个2048维向量，通过一个全连接然后用sigmoid函数得到整体图像一致性打分。

Type	Kernel	Stride	Outputs
conv.	5×5	2×2	64
conv.	5×5	2×2	128
conv.	5×5	2×2	256
conv.	5×5	2×2	512
conv.	5×5	2×2	512
FC	-	-	1024

表3 局部鉴别网络结构设置

Type	Kernel	Stride	Outputs
conv.	5×5	2×2	64
conv.	5×5	2×2	128
conv.	5×5	2×2	256
conv.	5×5	2×2	512
conv.	5×5	2×2	512
conv.	5×5	2×2	512
FC	-	-	1024

表4 全局鉴别网络结构设置

Type	Kernel	Stride	Outputs
concat.	-	-	2048
FC	-	-	1

表5 连接层设置

3.全局与局部一致性修补算法

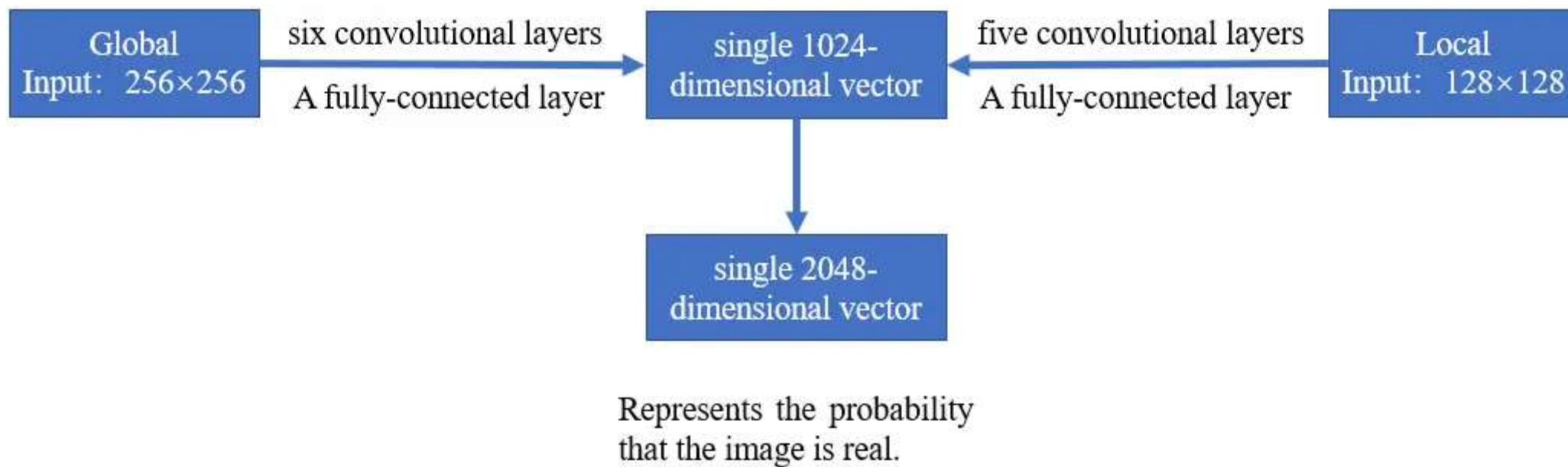


图5 鉴别器的工作示意图

3.全局与局部一致性修补算法

- 生成网络模型降低了两次分辨率，使用大小为原始大小四分之一的卷积，其目的是为了降低最终图像的纹理模糊程度。
- 文中使用了空洞卷积层（Dilated Convolution Layer），空洞卷积的优势在于它可以增加卷积层的感受野（receptive field）大小，同时不增加参数量。这使得网络可以捕捉更广阔的上下文信息，有助于处理具有大范围空间相关性的任务。如果感受野太小，那么在处理大的mask的时候很困难。

3.全局与局部一致性修补算法

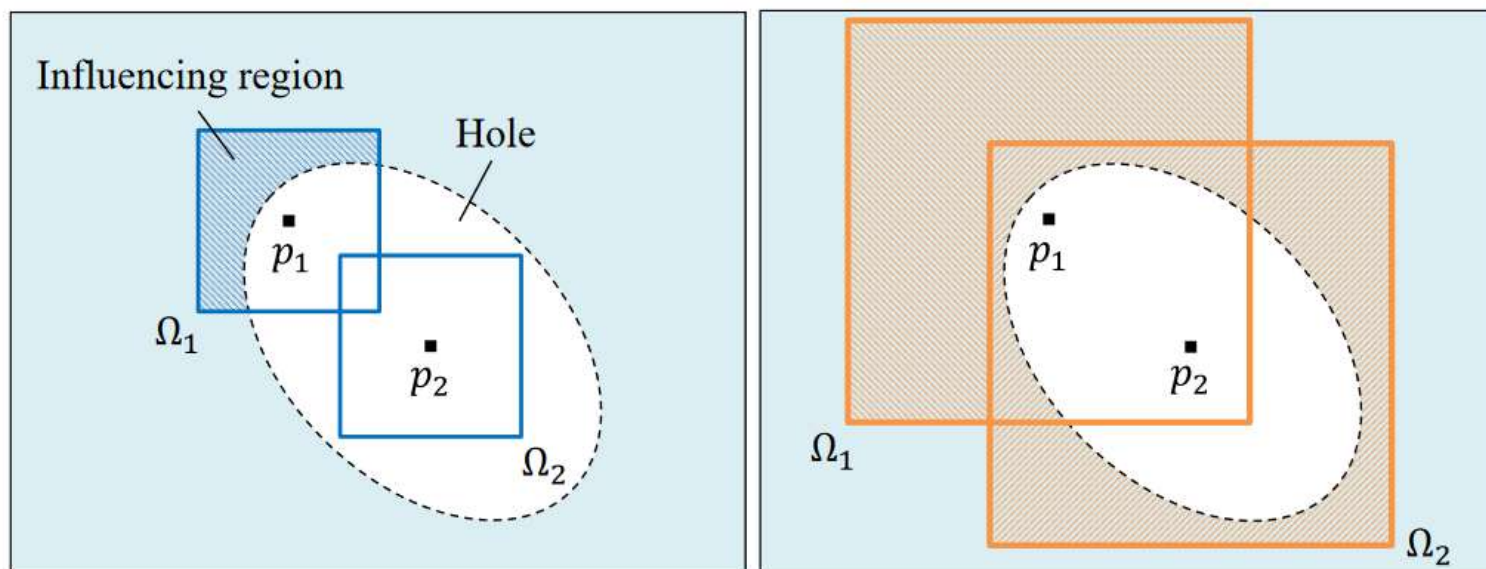


Fig. 3. Importance of spatial support. In order to be able to complete large regions, the spatial support used to compute an output pixel must include pixels outside of the hole. On the left, the pixel p_1 is computed from the influencing region in the spatial support Ω_1 , while the pixel p_2 cannot be calculated since the supporting area Ω_2 does not contain any information outside of the hole. However, on the right side, the spatial support is larger than the hole, allowing the completion of the center pixels.

3.全局与局部一致性修补算法

- **weighted Mean Squared Error (MSE)**作为损失函数

$$L(x, M_c) = \| M_c \odot (C(x, M_c) - x) \|^2$$

其中 \odot 表示逐像素相乘， $\|\cdot\|$ 是欧基里德范式。

- **GAN Loss**，目标是最大化生成图像和原始图像的相似概率

$$\min_C \max_D \mathbb{E}[\log D(x, M_d) + \log(1 - D(C(x, M_c), M_c))]$$

其中 M_d 是随机Mask， M_c 是输入Mask，期望值是所有训练图像的平均值。

3.全局与局部一致性修补算法

- 两个损失函数结合，得到总的损失函数：

$$\min_C \max_D \quad \mathbb{E}[L(x, M_c) + \alpha \log D(x, M_d) + \alpha \log(1 - D(C(x, M_c), M_c))]$$

3.全局与局部一致性修补算法



3.全局与局部一致性修补算法

[Huang et al. 2014]



[Pathak et al. 2016]



Ours



3.全局与局部一致性修补算法

不足:

- (1) 不能补全具有大尺寸洞的图，这主要与空洞卷积所能“看到”的区域大小有关，文中最大可看到 307×307 pixels。
- (2) 如果缺失区域在图像的边界，则处理不好。
- (3) 不能生成复杂性的结构性纹理，这个可能与作者的网络设计的网络有关。
- (4) 一定程度上受到训练图像库的限制，如果训练集中有一张和测试集相似图片，则效果会好一些。

3.全局与局部一致性修补算法

Input



[Pathak et al. 2016]



[Hays and Efros 2007]



Ours



失败案例图

提纲

- 智能图像修复
- 智能彩色图像处理

1、智能图像彩色化处理技术的发展

- 彩色图像与灰度图像之间的转换是一个不可逆的过程。
- 一幅灰度图像也可以对应多种彩色化方案，为彩色化处理问题的研究带来了一定复杂性。
- 对图像进行彩色化处理，一方面可以增加图像的真实感，提高图像的艺术性，也可以提高图像内容的可辨识度。
- 传统的图像伪彩色化方法主要有灰度分层法和灰度变换法。这些处理方法主要根据图像的局部灰度特征进行映射处理。其主要问题在于，对图像的灰度映射并没有考虑图像的语义信息，即不涉及对图像的理解。因此，传统的处理方法往往不能获得理想的处理效果。

1、智能图像彩色化处理技术的发展

■ 基于深度学习的彩色化处理技术

- ✓ 2015年，ICCV (International Conference on Computer Vision)上的论文发表了一篇关于深度学习的图像染色方法——Deep Colorization，该方法的特点是全自动上色，并且着色速度具有一定优越性。
- ✓ 2016年，ECCV (European Conference on Computer Vision)发表Colorful Image Colorization中提出了自监督学习的彩色化方法，解决了着色处理中的方案不确定问题，也给出了一种着色评估的有效策略。
- ✓ 2016年，ECCV ，Gustav Larsson, Michael Maire, and Gregory Shakhnarovich. Learning Representations for Automatic Colorization. In ECCV 2016.

1、智能图像彩色化处理技术的发展

- ✓ **Let there be Color!: Joint End-to-end Learning of Global and Local Image Priors for Automatic Image Colorization with Simultaneous Classification**; Satoshi Iizuka* Edgar Simo-Serra* Hiroshi Ishikawa（佐藤一裕*, 埃德加·西莫-索拉*, 石川博司），SIGGRAPH（Special Interest Group for Computer GRAPHICS，计算机图形图像特别兴趣小组，CCF推荐A类会议）2016
- ✓ 2017年，Unsupervised Diverse Colorization via Generative Adversarial Networks中采用条件生成网络结构进行探索，在每层特征中采用图像灰度信息作为彩色化的条件约束，取得了真实感的彩色化效果。同年，典型的彩色化处理方案Learning Diverse Image Colorization被提出，这个研究可以产生空域连续性的色彩效果，它采用自动变分编码器对低维度特征进行学习，建立与图像灰度之间的条件关系模型。

1、智能图像彩色化处理技术的发展

■ 基于深度学习的彩色化处理技术

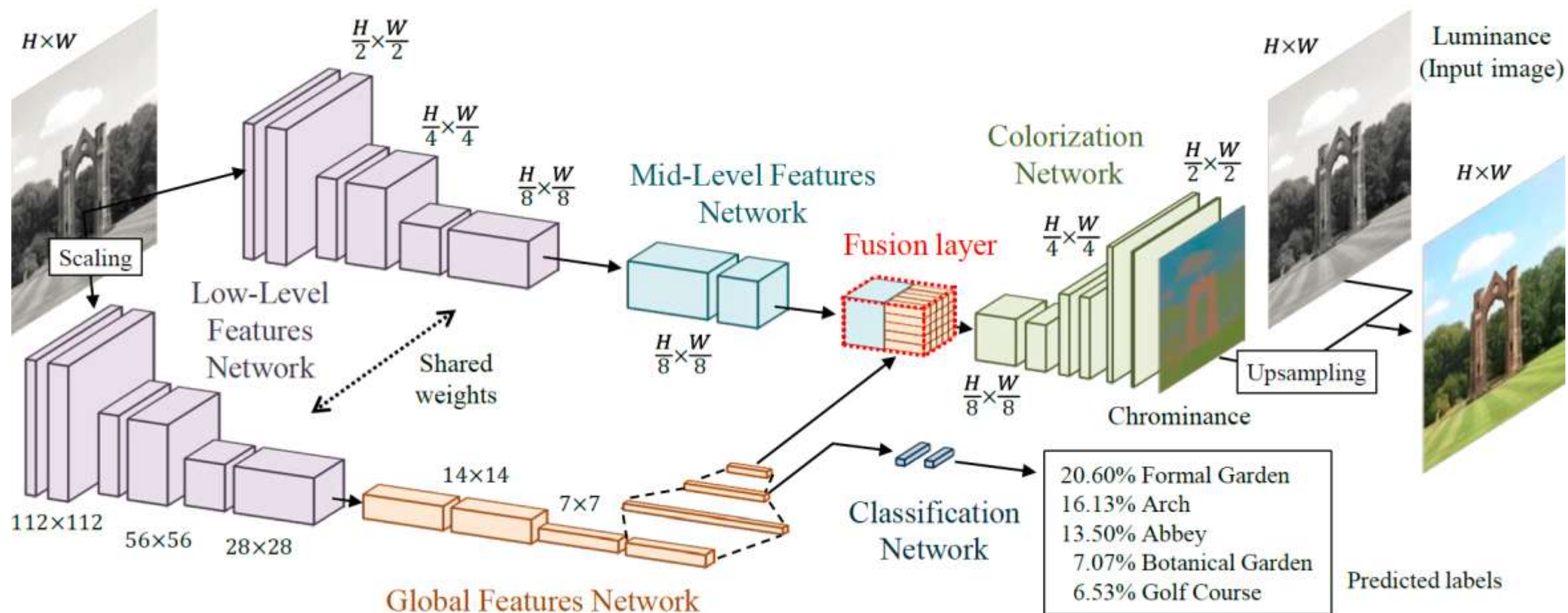
- ✓ 2018 ECCV, 针对图像结构不一致的问题, 在Structural Consistency and Controllability for Diverse Colorization中研究了通用结构的彩色化方法。
- ✓ CVPR 2019, 论文Coloring With Limited Data: Few-Shot Colorization via Memory Augmented Networks 中研究了小样本的彩色化方法。该方法在标签有限的情况下, 可以产生高质量的彩色化结果。同年, 也提出了一种利用自正则化手段产生全自动视频的着色方法Fully Automatic Video Colorization with Self-Regularization and Diversity。
- ✓ CVPR 2020, 基于实例感知的彩色化方法Instance-aware Image Colorization提出。该方法利用目标检测器获取目标图像, 使用实例着色网络来提取目标特征, 根据这些特征进行颜色生成。

1、智能图像彩色化处理技术的发展

■ 基于深度学习的彩色化处理技术

- ✓ ICIR (International Conference on Learning Representations) 2021, 一种基于自注意力的方法Colorization Transformer被提出, 其使用Transfomer产生灰度图像的低分辨率粗糙色彩, 进一步采用条件Tansfommer层来有效调节灰度输入学习彩色化的特征。
- ✓ 总体来看, 基于深度学习的彩色图像处理技术主要有全自动图像着色、无需用户进行交互的全自动的彩色化方法和用户引导的策略。在用户引导的技术研究中, 出现了根据用户交互引导的方法、用户指定颜色板的引导策略、用户提供参考色图像的彩色化处理、用户指定色块的彩色化处理以及基于自然语言描述引导的彩色化措施。其中, 用户引导的上色和基于样本的上色是两种常见的用户引导的彩色化策略。

Let there be Color!



Let there be Color!

提出了一种结合全局先验和局部图像特征的新方法，用于自动为灰度图像着色。基于卷积神经网络提出的深度网络包含一个融合层，可以优雅地将依赖于小图像块的局部信息与使用整个图像计算的全局先验相结合。整个框架，包括全局和局部先验以及着色模型，均以端到端的方式进行训练。

与大多数基于CNN的现有方法不同，论文提出的架构可以处理任意分辨率的图像。利用现有的大规模场景分类数据库来训练模型，通过使用数据集的**类别标签**，更高效和更具区分性地学习**全局先验**。论文的方法在许多不同类型的图像上广泛进行了验证，包括百年前的黑白摄影，结果显示了逼真的着色效果。

<https://iizuka.cs.tsukuba.ac.jp/projects/colorization/en/>

Let there be Color!

Colorization of Black-and-white Pictures



Let there be Color!

Our Goal: Fully-automatic colorization



2. YUV空间彩色化处理

Cheng Z, Yang Q, Sheng B. [Deep Colorization](#)[C]// IEEE International Conference on Computer Vision. 2015.

论文贡献

- 提出一种全自动着色方法，它利用来自不同类别（如动物、户外、室内）的**大量参考图像**和各种对象（如树、人、熊猫、汽车等）。
- 使用深度神经网络，**利用YUV色彩空间**
- 详细分析了信息量大、鉴别能力强的图像特征描述子，得出结论：选择合适的算子是论文中图像彩色化方法成功实现的关键。

2. YUV空间彩色化处理

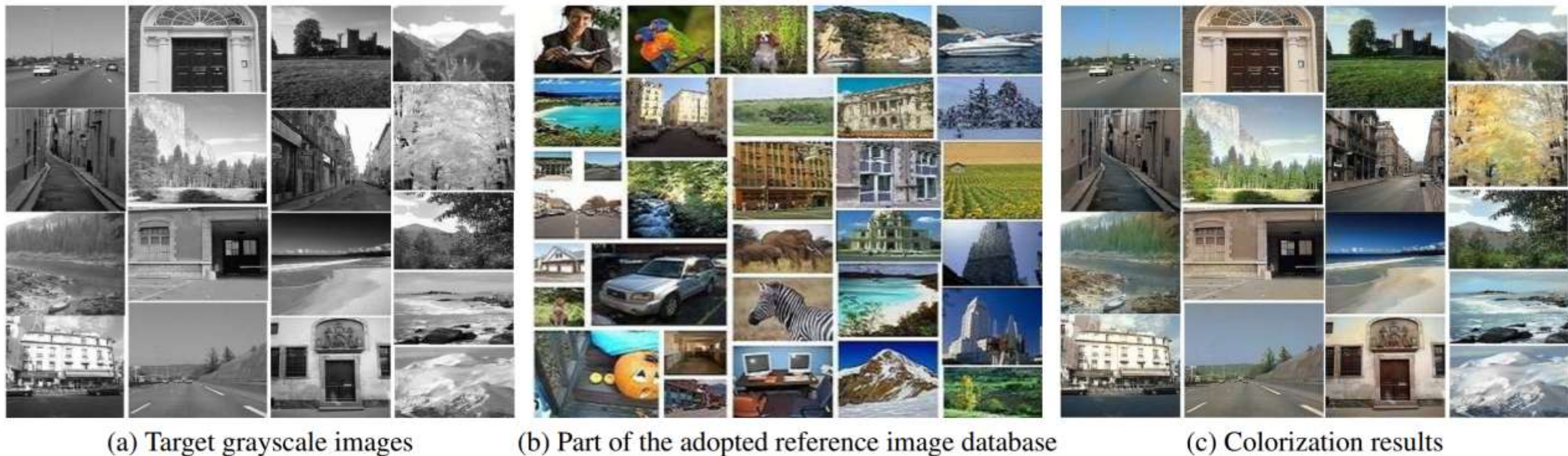


Figure 1. The colorization results of our full-automatic method. Our system utilizes a large database of colorful reference images as shown in (b). After the training of neural networks, the learned model is used directly to colorize the target gray scale images in (a). The colorization results are presented in (c).

2. YUV空间彩色化处理

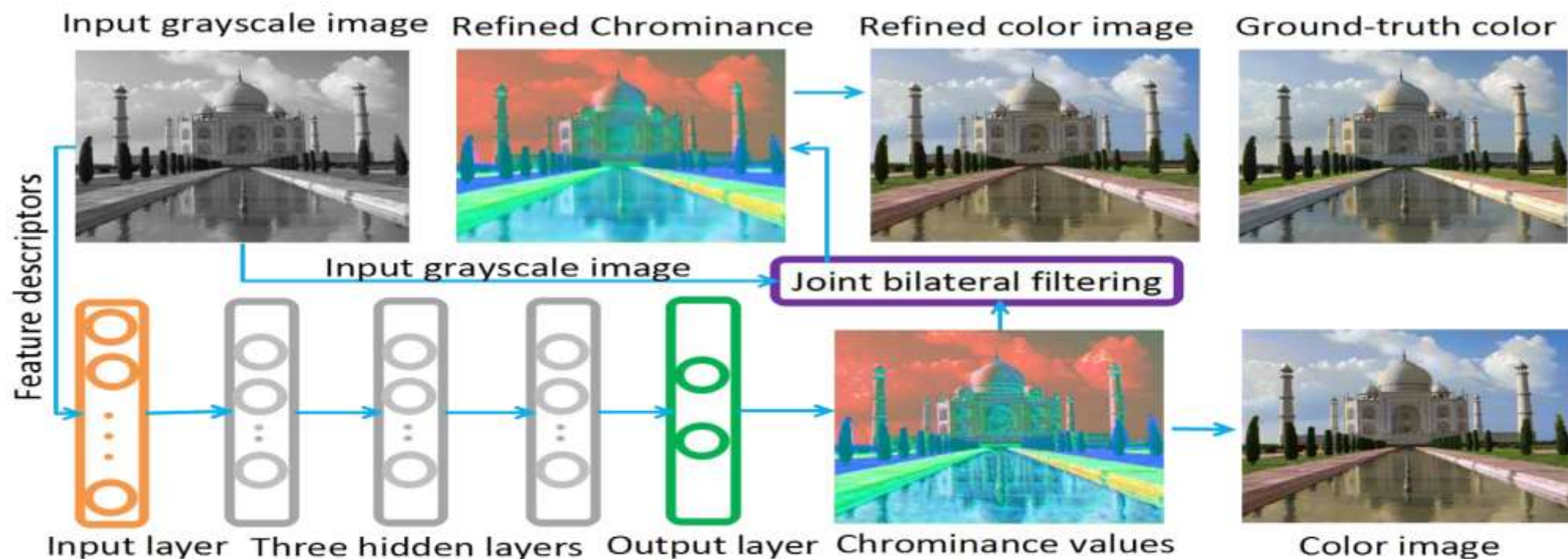


Figure 2. Overview of the proposed colorization method and the architecture of the adopted deep neural network. The feature descriptors will be extracted at each pixel and serve as the input of the neural network. Each connection between pairs of neurons is associated with a weight to be learned from a large reference image database. The output is the chrominance of the corresponding pixel which can be directly combined with the luminance (grayscale pixel value) to obtain the corresponding color value. The chrominance computed from the trained model is likely to be a bit noisy around low-texture regions. The noise can be significantly reduced with a joint bilateral filter (with the input grayscale image as the guidance).

2. YUV空间彩色化处理

网络结构

- 深度神经网络由一个输入层、多个隐藏层和一个输出层组成。每层可包含不同数量的神经元。
- 输入层的神经元的数量等于（从灰度图像）提取特征的维数，具体包含有128个神经元，对应49维的低级特征（patch feature）、32维的中级特征（DAISY feature）和47维的高级语义特征。
- 输出相应颜色值的U和V通道
- 隐层神经元个数为输入的一半
- 隐藏层或输出层中的每个神经元都与下一层的所有神经元相连，并且每个连接都具有一个相关权重。

2. YUV空间彩色化处理

两个主要步骤:

- 训练: 使用大规模图像数据集训练神经网络;
- 预测: 使用学习的神经网络对目标灰度图像进行着色。

Algorithm 1 Image Colorization – Training Step

Input: Pairs of reference images: $\Lambda = \{\vec{G}, \vec{C}\}$.

Output: A trained neural network.

1. Compute feature descriptors \vec{x} at sampled pixels in \vec{G} and the corresponding chrominance values \vec{y} in \vec{C} ;
2. Construct a deep neural network;
3. Train the deep neural network using the training set $\Psi = \{\vec{x}, \vec{y}\}$.

Algorithm 2 Image Colorization – Testing Step

Input: A target grayscale image I and the trained neural network.

Output: A corresponding color image: \hat{I} .

1. Extract a feature descriptor at each pixel location in I ;
2. Send feature descriptors extracted from I to the trained neural network to obtain the corresponding chrominance values;
3. Refine the chrominance values to remove potential artifacts;
4. Combine the refined chrominance values and I to obtain the color image \hat{I} .

2. YUV空间彩色化处理

- 将图像着色问题看成一个回归问题，并采用一般的神经网络结构来求解。
- 对有监督的学习样例： $\Lambda = \{\vec{G}, \vec{C}\}$ ，彩色化处理的目标就是求取一个复杂的映射函数 \mathcal{F} ，该函数可以通过特征学习，实现将灰度图像 \vec{G} 中的每个灰度值映射到相应的 \vec{C} 中的色度值。 \vec{G} 表示灰度图像， \vec{C} 表示彩色图像。

2. YUV空间彩色化处理

算法的目标是从 Λ 学习一个映射函数，这样就可以使用 \mathcal{F} 将一个新的灰色图像转换为彩色图像。

论文使用了YUV颜色空间，因为这个颜色空间**最小化了颜色空间三个坐标轴之间的相关性**。对于 \vec{G} 中的像素 p ， \mathcal{F} 的输出只是 \vec{C} 中对应像素的U和V通道， \mathcal{F} 的输入是像素 p 计算的特征描述符。论文将灰度到颜色的映射函数重新表述为 $c_p = \mathcal{F}(\Theta, x_p)$ ，其中 x_p 是在像素 p 处提取的特征描述符， c_p 是对应的色度值。 Θ 是要从 Λ 中学习的映射函数 \mathcal{F} 的参数。

2. YUV空间彩色化处理

通过求解最小二乘最小化问题来学习参数

$$\operatorname{argmin}_{\Theta \in \Upsilon} \sum_{p=1}^n \| \mathcal{F}(\Theta, x_p) - c_p \|^2$$

其中 n 是从 Λ 中取样的训练像素总数， Υ 是 $\mathcal{F}(\Theta, x_p)$ 的函数空间。

2. YUV空间彩色化处理

特征描述符

特征设计是实现彩色化的关键。论文进行了大量的实验来测试各种特征，并且只保留对着色结果有实际影响的特征。进而将采用的特性分为低、中、高级别特性。 $\mathbf{x}_p = \{\mathbf{x}_p^L; \mathbf{x}_p^M; \mathbf{x}_p^H\}$

- Low-level Patch Feature: 以 p 为中心， 7×7 的邻域
- Mid-level DAISY Feature: DAISY是一个用于密集匹配的快速局部描述符。与底层块特征不同，DAISY可以对局部块进行更准确的识别描述，从而提高复杂场景下的着色质量。
- High-level Semantic Feature: 场景分析算法对每个像素进行分类标注，得到输入图像的语义映射。每个像素位置计算一个一维概率向量，其中 N 是对象类别的总数。每个元素都是当前像素属于相应类别的概率。该概率向量用作表示为 \mathbf{x}^H 的高级描述符。

2. YUV空间彩色化处理

训练采用Sun database，2688张图像进行训练；每个图像被分割成若干个对象区域，共使用47个对象类别（例如建筑、汽车、海洋等）。

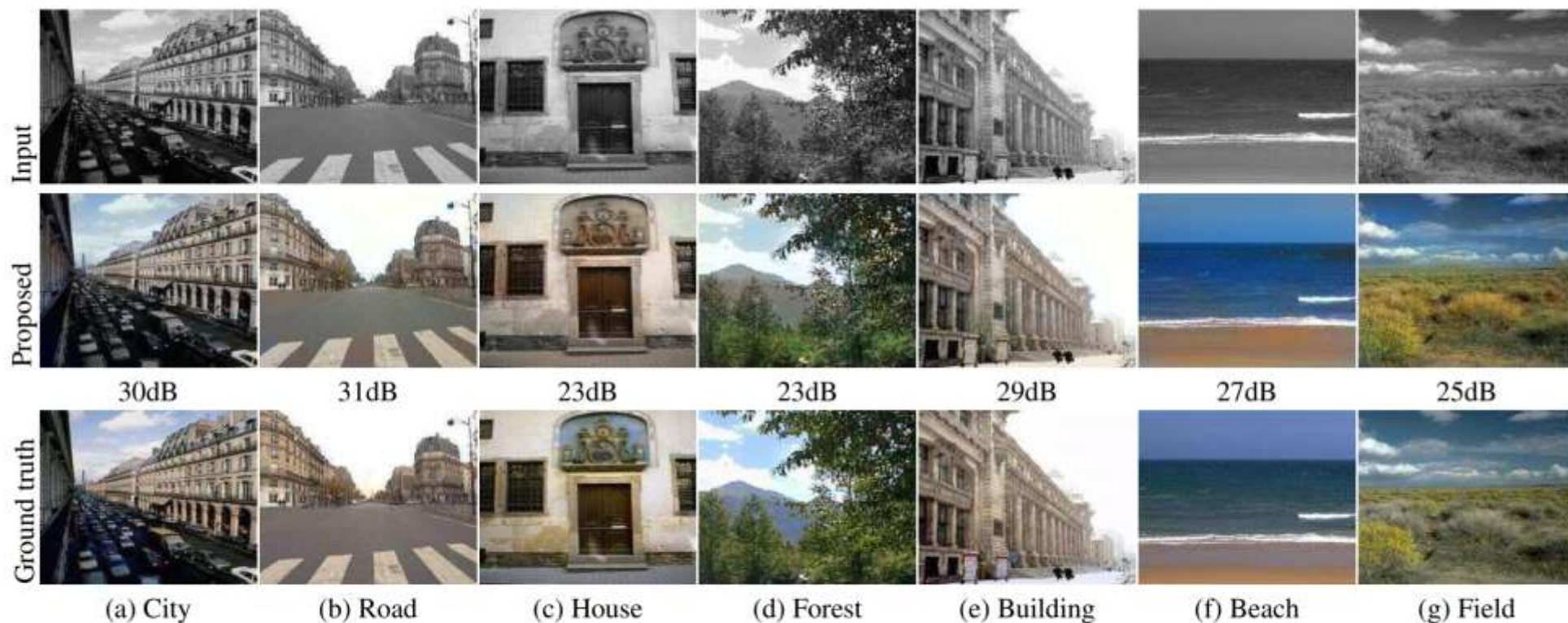


Figure 9. Comparison with the ground truth. The first row presents the input grayscale images from different categories. Colorization results obtained from the proposed method are presented in the second row. The last row presents the corresponding ground-truth color images, and the PSNR values computed from the colorization results and the ground truth are presented under the colorized images.

2. YUV空间彩色化处理

不足之处:

- 利用该方法对图像进行着色能够得到**基本正确的结果**，但是产生的颜色饱和度偏低，主要原因是由于损失的控制中采用预测结果和真实图像之间的欧式距离导致的。
- 它依赖于机器学习技术，并有其自身的局限性。例如，它需要在一个包含所有可能对象的大型参考图像数据库上进行训练。这在实践中是不可能的。例如，当前模型是在真实图像上训练时，对合成图像彩色化无效。

3. Lab空间彩色化处理

Zhang R, Isola P, Efros A A. Colorful Image Colorization [C]. ECCV , 2016

论文贡献

- 克服了对用户交互的依赖和不饱和着色问题
- 实现了自动图像着色，维持了颜色的多样性，可以产生生动逼真的着色
- 将图像着色任务转化为一个**自监督表达学习的任务**，取得了满意的结果
- 用灰度图像中物体的纹理、语义等信息作为线索，来预测真实感着色

3. Lab空间彩色化处理

■ 问题描述和背景

作者指出，在许多情况下，场景的语义和纹理信息提供了有关颜色的线索。例如，草地通常是绿色的，天空通常是蓝色的。因此，研究目标不是还原精确的真实颜色，而是生成一个能愚弄人类观察者的合理上色效果。

■ 解决方法

提出了一种全自动的方法，将其视为一个分类任务，通过训练数据来捕捉语义和纹理之间的统计依赖关系，从而生成生动和逼真的彩色图像。作者的方法基于CIE Lab颜色空间中的亮度通道L，将L通道输入到模型中，然后预测a和b颜色通道。训练数据采用大规模的彩色照片，使其具有充足的颜色信息。最后将原始的L通道和推理出的ab通道结合起来，得到一张上色的图片。

3. Lab空间彩色化处理

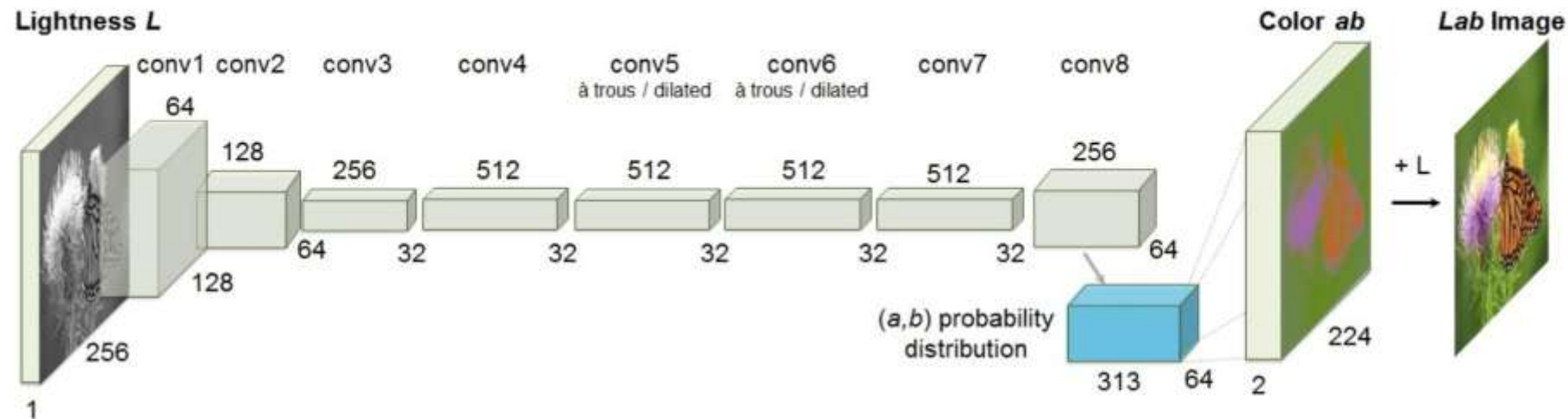


Fig. 2. Our network architecture. Each conv layer refers to a block of 2 or 3 repeated conv and ReLU layers, followed by a BatchNorm [30] layer. The net has no pool layers. All changes in resolution are achieved through spatial downsampling or upsampling between conv blocks.

3. Lab空间彩色化处理

- 目标是将灰度输入通道映 $\mathbf{X} \in \mathbb{R}^{H \times W \times 1}$ 映射到关联的两个颜色通道 \mathbf{Y} ，即学习关联映射 $\hat{\mathbf{Y}} = \mathcal{F}(\mathbf{X})$ 。 $\mathbf{Y} \in \mathbb{R}^{H \times W \times 2}$
- 为了实现这一任务，作者采用CIE Lab色彩空间，并使用欧氏损失函数来度量预测颜色和真实颜色之间的距离。
- 然而，由于颜色上色问题具有固有的多模态性和歧义性，欧氏损失函数不够鲁棒，容易产生灰暗和褪色的结果。因此，作者将问题视为多项式分类问题，将ab输出空间分成网格大小为10的区域，学习将输入 \mathbf{X} 映射为可能颜色的概率分布。

3. Lab空间彩色化处理

■ 损失函数和多模态性

设计了专门针对上色问题的损失函数。由于颜色预测是多模态的，即多个颜色可能对一个对象是合理的，作者采用了一种损失函数，鼓励多种可能颜色的预测，并在训练时重新加权以强调稀有的颜色。最终的上色效果是基于颜色分布的平均值，使得结果更加生动和真实。

■ 评估方法

为了评估上色结果，作者引入了一种“上色图灵测试”，通过人类参与者的选择来测试合成的彩色图像的视觉逼真度。结果显示，作者的算法成功地愚弄了人类观察者，并产生了接近照片质量的彩色图像。此外，论文还展示了生成的彩色图像在对象分类等下游任务中的实用性。

3. Lab空间彩色化处理

(1) 根据 \mathbf{X} , 预测向量 $\hat{\mathbf{Z}}$

For a given input \mathbf{X} , we learn a mapping $\hat{\mathbf{Z}} = \mathcal{G}(\mathbf{X})$ to a probability distribution over possible colors $\hat{\mathbf{Z}} \in [0, 1]^{H \times W \times Q}$ where Q is the number of quantized *ab* values.

(2) 监督数据的转换

To compare predicted $\hat{\mathbf{Z}}$ against ground truth, define function $\mathbf{Z} = \mathcal{H}_{gt}^{-1}(\mathbf{Y})$ which converts ground truth color \mathbf{Y} to vector \mathbf{Z} , using a soft-encoding scheme^[2].

(3) 使用交叉熵损失进行控制

then use multinomial cross entropy loss $L_{cl}(\cdot, \cdot)$, defined as:

$$L_{cl}(\hat{\mathbf{Z}}, \mathbf{Z}) = - \sum_{h,w} v(\mathbf{Z}_{h,w}) \sum_q \mathbf{Z}_{h,w,q} \log(\hat{\mathbf{Z}}_{h,w,q})$$

多项式交叉熵损失

当损失达到最小以后, 将预测的 $\hat{\mathbf{Z}}$ 转换为颜色结果:

Finally, we map probability distribution $\hat{\mathbf{Z}}$ to color values $\hat{\mathbf{Y}}$ with function $\hat{\mathbf{Y}} = \mathcal{H}(\hat{\mathbf{Z}})$

3. Lab空间彩色化处理

损失设计思路：设计损失函数、加权平滑像素损失、概率分布转点估计。

(1) 损失函数设计

- 首先将ab通道的输出空间以10为步长量化为 $Q=313$ 类（两个颜色通道ab值一起预测）
- 计算损失时将标签色值也转换到313范围的概率空间中

3. Lab空间彩色化处理

(2) 加权平滑像素损失

- 使用一种加权平滑像素损失的方式，训练时为每个像素的loss重新调整权重，权重大小基于像素颜色的分布，通过统计ImageNet训练集的色彩概率分布获取颜色分布。作者在1.3M张图片数据上统计发现，**低饱和度的颜色数量非常大**，如果忽视这种情况，训练的模型的输出依然是低饱和的图片。因此需要通过权重来使得类尽可能的均衡。权重的计算方法是：统计训练图片a，b量化后的分布（只取距离最近的），对分布使用高斯函数进行平滑，平滑后的分布的倒数即可作为权重。这里作者在加权重和不加权重上做了个折中，可以理解为最后的损失有两部分，一部分是没有乘权重的，另一部分是乘了权重的。论文通过在计算的权重上加一个均匀分布来实现这种折中。
- 使用类似代价敏感策略：对于越稀有色素给予更大的权重和关注力，获取的着色效果。

3. Lab空间彩色化处理

(3) 概率分布转点估计

- 通过重新调整Softmax分布的温度T，并取结果的平均值进行插值

模型对于每个位置输出313个可能颜色的概率，用所得的概率对313个区间的中心加权求和可得到最终的颜色。论文中通过实验发现，如果直接用模型输出的概率计算颜色，颜色饱和度会比较低；如果只选取最大的概率的区间所表示的颜色，整个图片会不连续。作者通过下面的公式来调整模型输出的概率，然后用改变后的概率对区间中心加权求和得到最终的颜色， $T=0.38$ 。

$$H(Z_{h,w} = E[f_T(Z_{h,w})]) \quad f_T(z) = \frac{\exp(\log(z)/T)}{\sum_q \exp(\log(z_q)/T)}$$

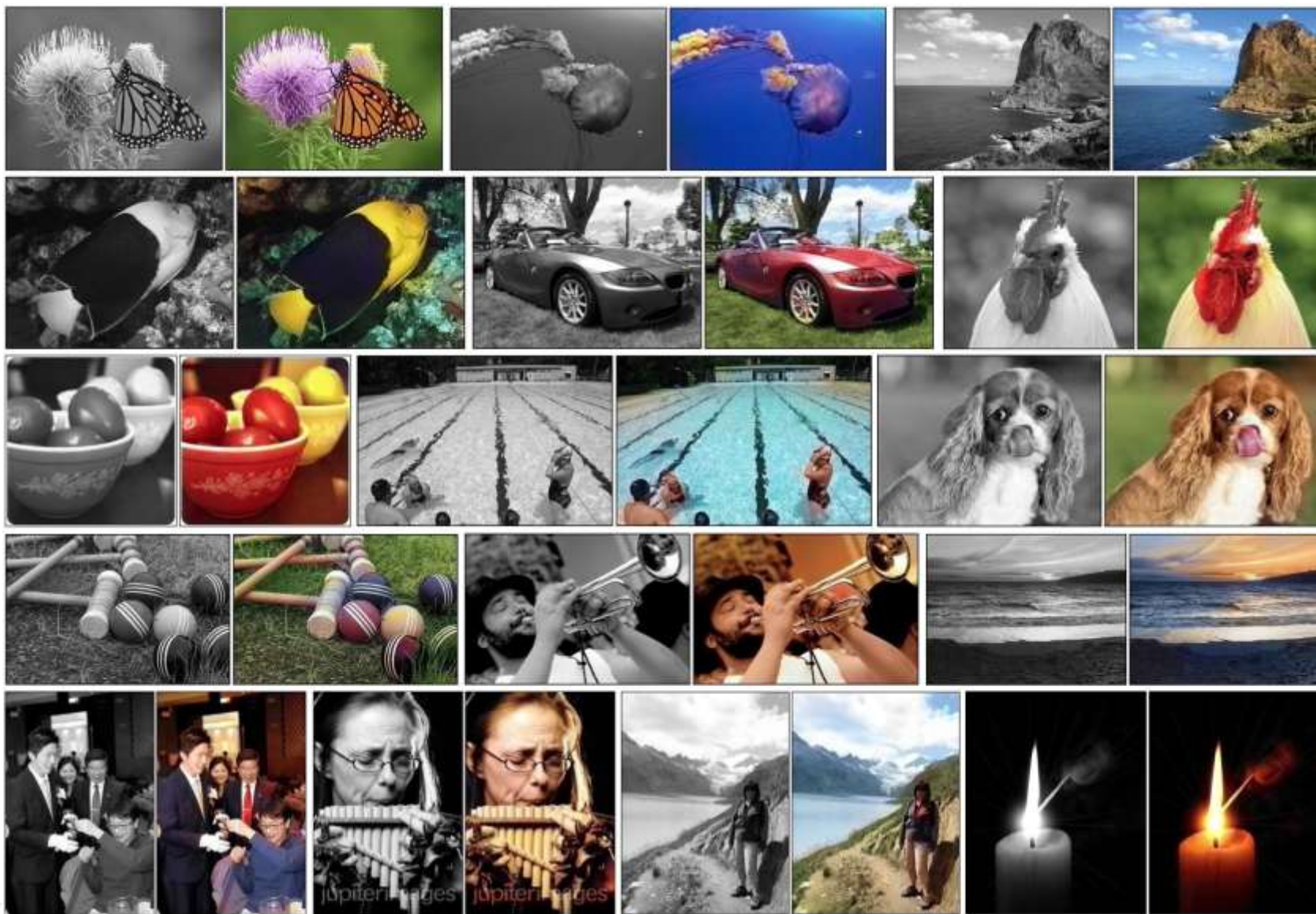
3. Lab空间彩色化处理

训练

对来自ImageNet训练集：

- 1.3M图像进行网络训练
- 对ImageNet验证集中的前10K图像进行验证
- 对验证集中的单独10K图像进行测试
- 所选成功和失败案例的定性比较如图所示。

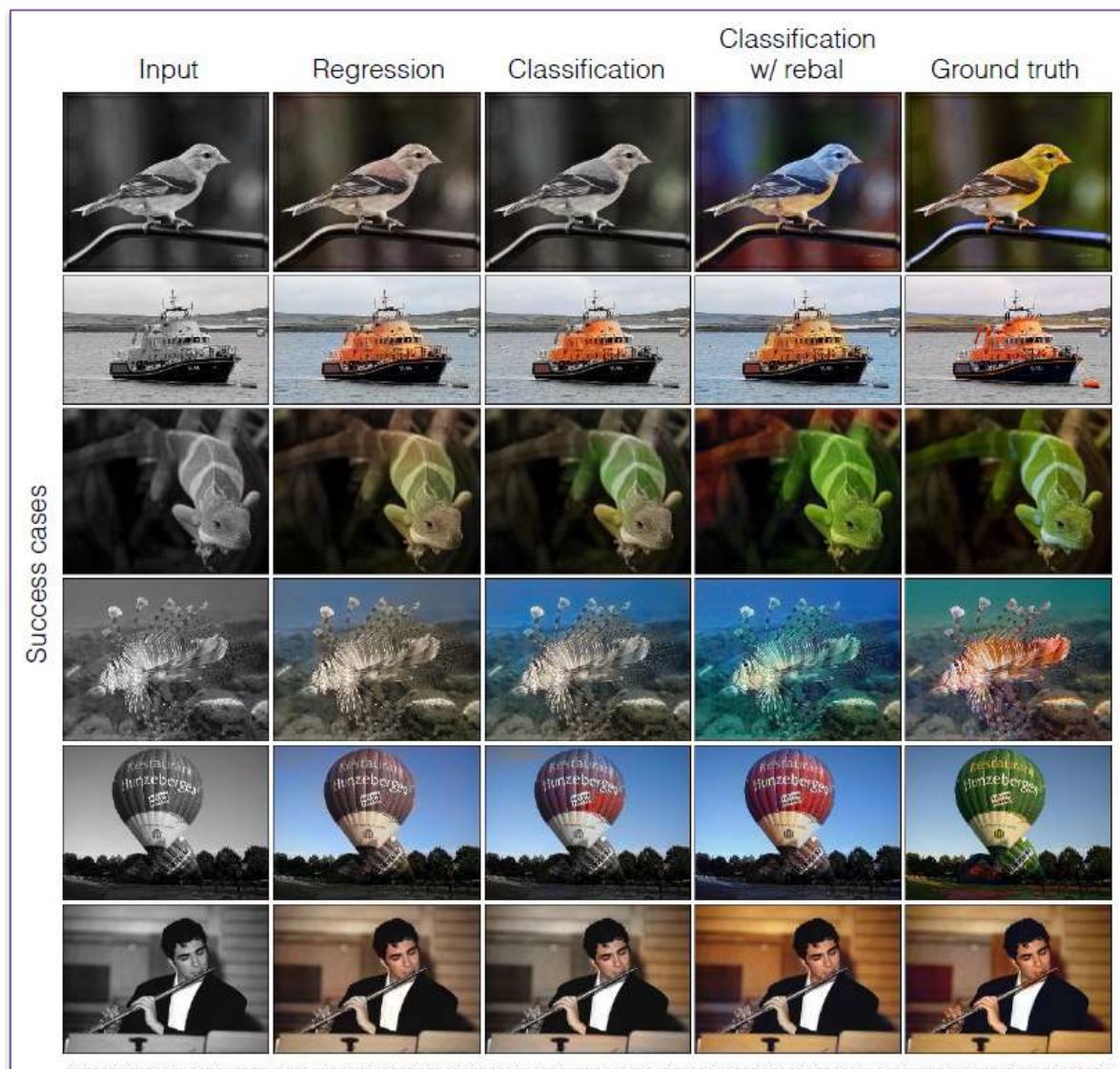
3. Lab空间彩色化处理



especially well results

3. Lab空间彩色化处理

成功结果



3. Lab空间彩色化处理

失败结果



THE END
Thank You.

