Pik-Fix: Restoring and Colorizing Old Photos

Runsheng Xu[1†] Zhengzhong Tu[2†] Yuanqi Du[3†] Xiaoyu Dong[4] Jinlong Li[5] Zibo Meng[6]
Jiaqi Ma[1*] Alan Bovik[2] Hongkai Yu[5*]
[1] UCLA [2] UT-Austin [3] George Mason University [4] Northwestern University
[5] Cleveland State University [6] Innopeak Technology Inc.

CVPR 2022

2022.05.16
Presenter: Sohee Jeong

Contents

Introduction

Motivation

- Old photo often requires both restoration and colorization simultaneously
- Lack of large-scale dataset of old photos makes restoration task very challenging

Contribution

- Reference-based end-to-end deep learning framework
- Learns to simultaneously restore and colorize the old photos
- Created a first-of-a-kind public dataset

Related work

1. Image Colorization

What is Image Colorization?

- Grayscale image → color image
- Ill-posed problem: no unique color

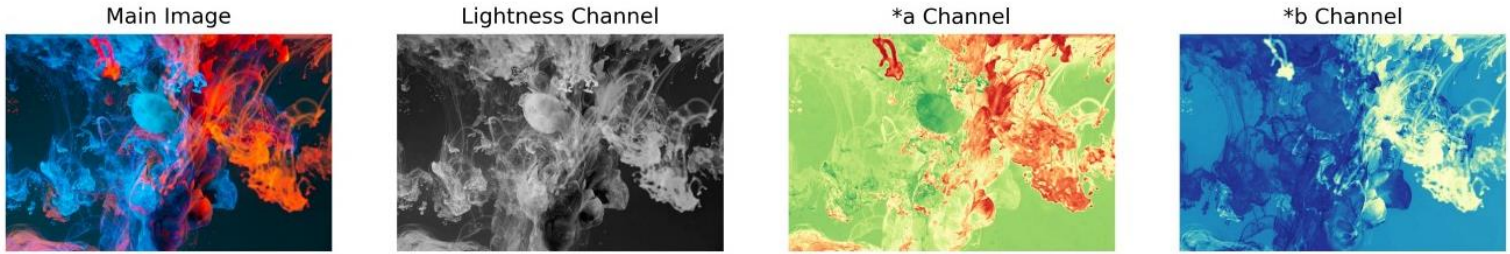
Related work

CIELAB Color Space

- Separate the luminance component from the color information
- Perceptually uniform color space
- Device-independent

Why use Lab rather than RGB?

- To reduce the complexity of the task
- predict 3 information R,G,B → 2 information a,b



Related work

1) Scribble-based method

- User-made scribbles are placed upon certain areas of the image
- Assumption of spatial continuity
- Pros: no need for searching reference image, user can adjust color
- Cons: requires human effort, highly depends on user selection

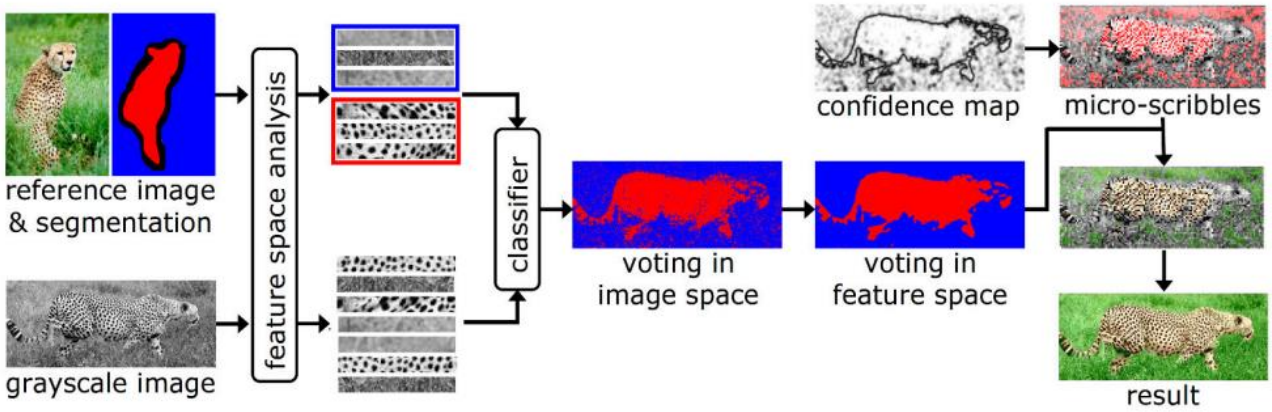
(a) (b) (c)

FIGURE 1. a) Original photograph, b) scribbles applied on the grayscale version of the photograph, c) colorization results.

Related work

2) Example-based method

- Transfer color information from **color source** image → matching regions of **target grayscale** image
- Correspondence between the input pictures and the reference data



R. Irony, D. Cohen-Or, and D. Lischinski, ‘‘Colorization by example,’’ in Proc. Eurographics Symp. Rendering, Konstanz, Germany, 2005, pp. 201–210.

Related work

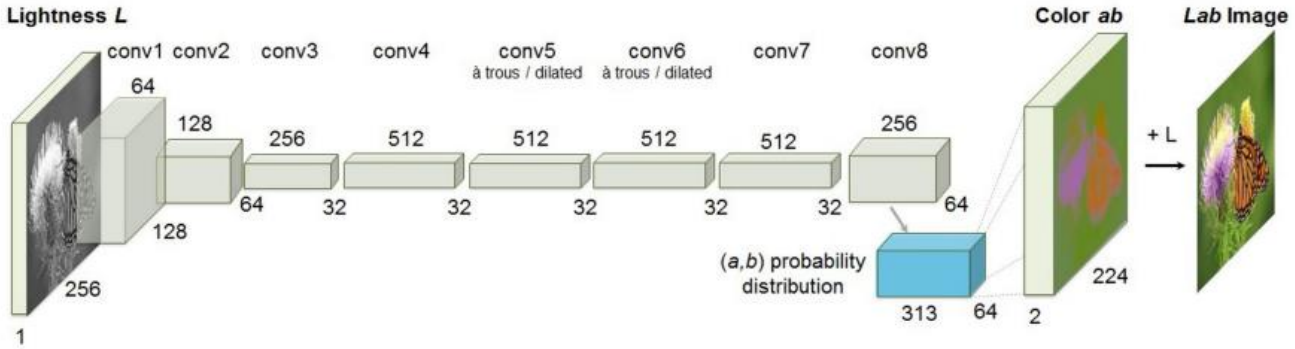
1. Image Colorization

- Pros
  - ✓ Simplicity, speed
- Cons
  - ✓ possible non-existence of suitable reference image
  - ✓ result is highly dependent on the quality of the reference image
  - ✓ reference need to implicate visual similarity

Related work

3) Deep-learning based

- Model automatically learns to map colors that naturally correspond to real objects
- Pros: reducing user effort
- Cons: numerous parameters need to be tuned, large dataset needed, time consuming



R. Zhang, P. Isola, and A. A. Efros, ‘‘Colorful image colorization,’’ in Proc. Eur. Conf. Comput. Vis., 2016, pp. 649–666

Related work

2. Image Restoration

Degradations

- Capture defects: blur, exposure
- Physical defects: cracks, tears, smudges
- Multiple picture degradations exist simultaneously

Noise Low-resolution Blur

Old photo restoration

- Often requires both distortion restoration and colorization
- Degradation ↔ Colorization

Methodology

Overall Flow

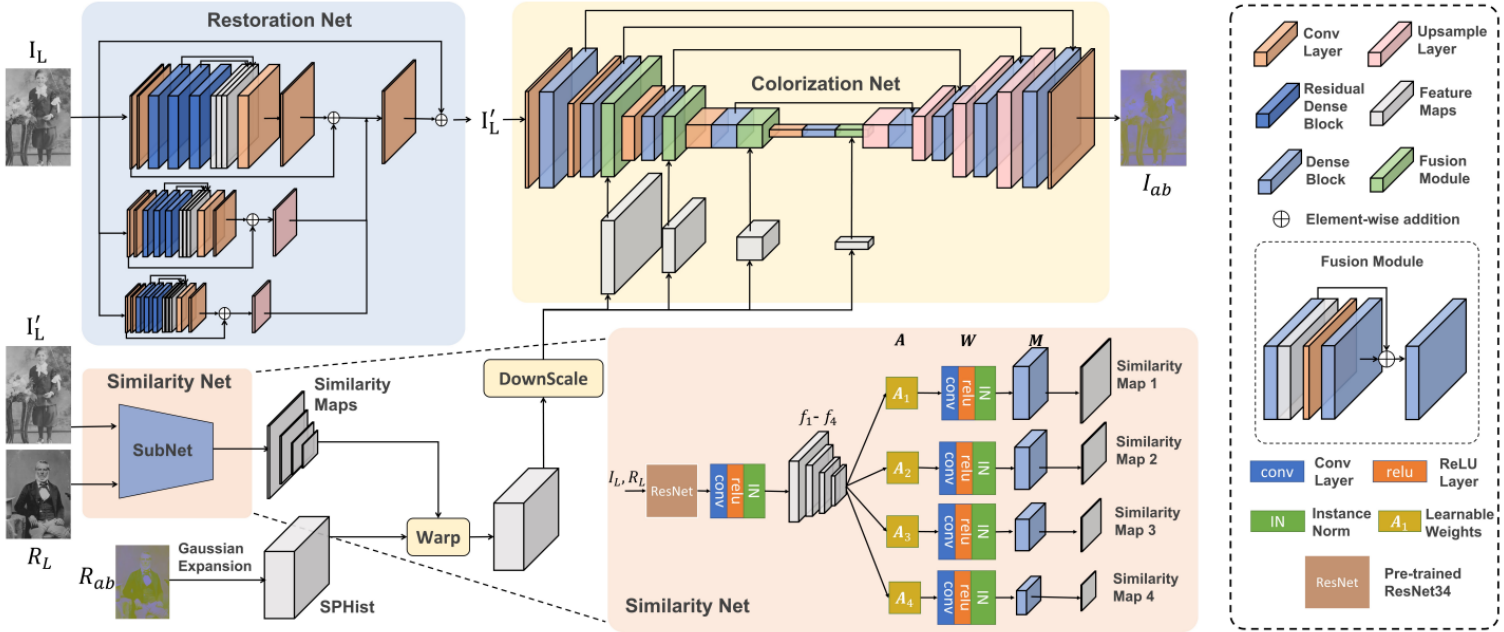
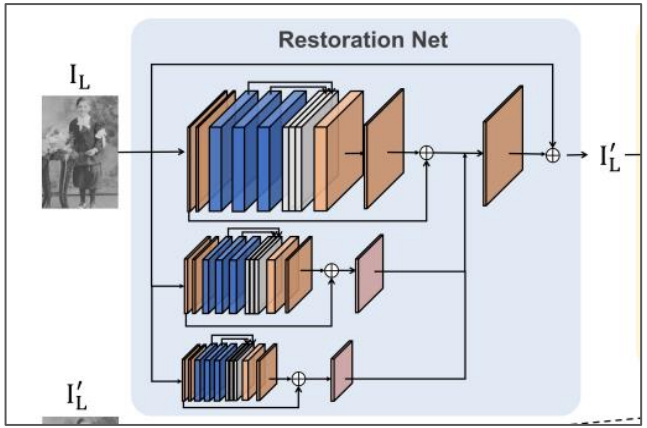
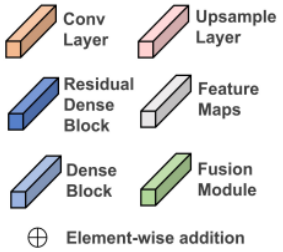


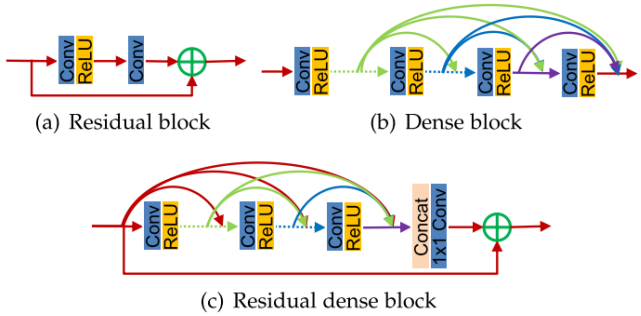
Figure 2: Flow diagram of the triplet networks (restoration, similarity and colorization) that define the flow of visual information processing in Pik-Fix.

Methodology

1. Restoration Sub-Net







$I_L \in \mathbb{R}^{H\times W\times 1}, I'_L \in R^{H\times W\times 1}$

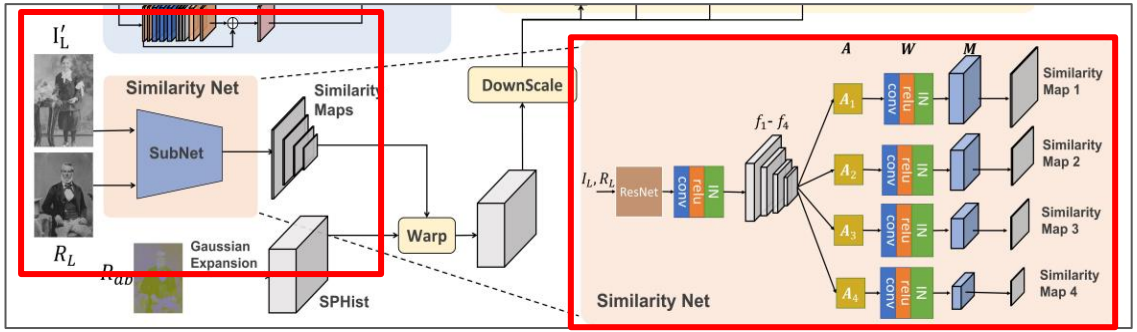
$R_L \in \mathbb{R}^{H\times \mathrm{W}\times 1}, R_{ab} \in \mathbb{R}^{H\times W\times 2}$

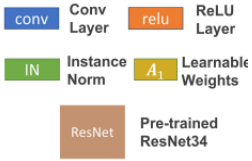
$I_{Lab} \in \mathbb{R}^{H\times W\times 3}$

- Multi-level Residual Dense Network (RDN)
- Enlarge receptive field to handle broader range of distortions
- $1\times, 4\times, 8\times$ downsampled input is fed into top, second, third level of the RDN respectively
- Generate the restored luminance I'_L

Methodology

2. Similarity Sub-Net





$*$: element-wise multiplication
$\circledast$: convolution operator
$\bigoplus$: concatenation

- Feature map $f_i \in \mathbb{R}^{H_i \times W_i \times C}$ $(i = 1,2,3,4)$, Learnable coefficient $A_i \in \mathbb{R}^{1\times 4}$ $(\mathrm{i} = 1,2,3,4)$
- Concatenated feature $M_i \in \mathbb{R}^{H_i \times W_i \times C}$,
  $M_i = W \circledast [g(A_{i1} * f_1) \bigoplus g(A_{i2} * f_2) \bigoplus g(A_{i3} * f_3) \bigoplus g(A_{i4} * f_4)]$, $\overline{M_i} \in \mathbb{R}^{H_i W_i \times C}$
- **Similarity map $\boldsymbol{\Phi^i_{R\leftrightarrow I}} \in \mathbb{R}^{H_i W_i \times H_i W_i}$**

$$\Phi^i_{R\leftrightarrow I}(u,v) = \frac{(\bar{M_i}^I(u) - \mu_{\bar{M_i}^I}) \cdot (\bar{M_i}^R(v) - \mu_{\bar{M_i}^R})}{||\bar{M_i}^I(u) - \mu_{\bar{M_i}}||_2 ||\bar{M_i}^R(u) - \mu_{\bar{M_i}^R}||_2},$$ at each spatial location (u, v)

Methodology

3. Colorization Sub-Net

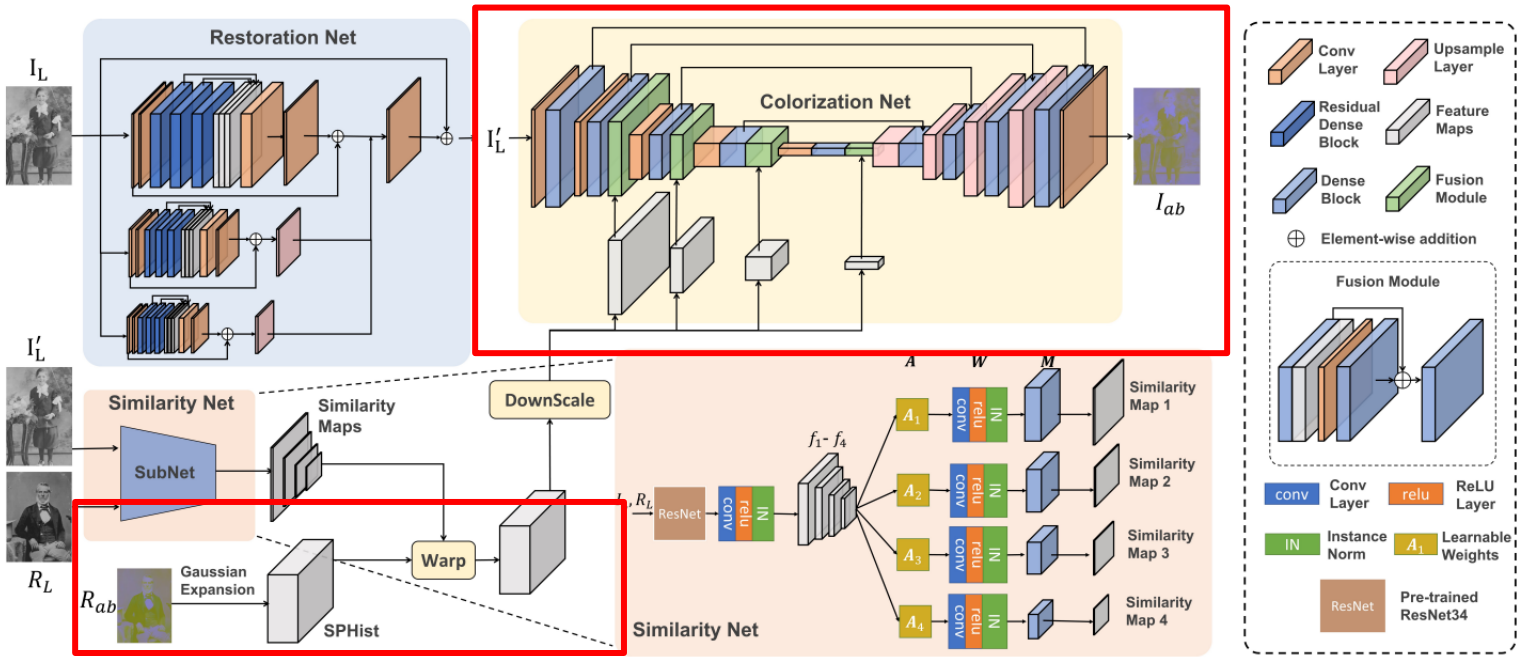
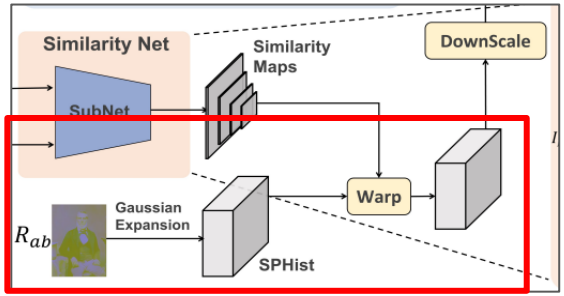


Figure 2: Flow diagram of the triplet networks (restoration, similarity and colorization) that define the flow of visual information processing in Pik-Fix.

Methodology

3. Colorization Sub-Net

1. Approximate **SPHist** $h \in \mathbb{R}^{H \times W \times K}$ for each channel

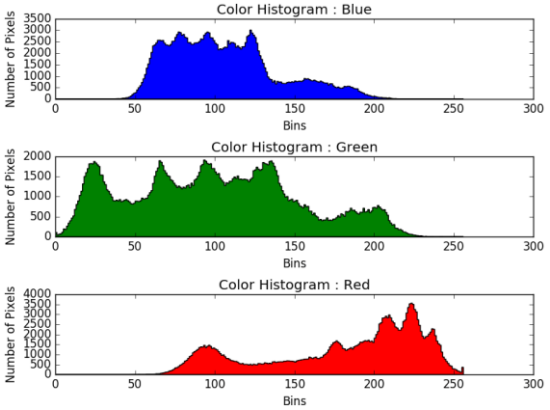


SPHist (Space-preserving color histogram)

- Retain spatial picture information
- Probability of pixel at location (i, j) falling into the k^{th} bins:

$$h(i,j,k) = \frac{\exp(-(D_{ij} - u_k)^2 / 2\sigma^2)}{\sum_{k=1}^{K} \exp(-(D_{ij} - u_k)^2 / 2\sigma^2)},$$

K: number of histogram bins
D_{ij}: value of a,b channel of the reference picture at (i, j)
σ=0.1, μ_k: center of bin k. learnable parameter



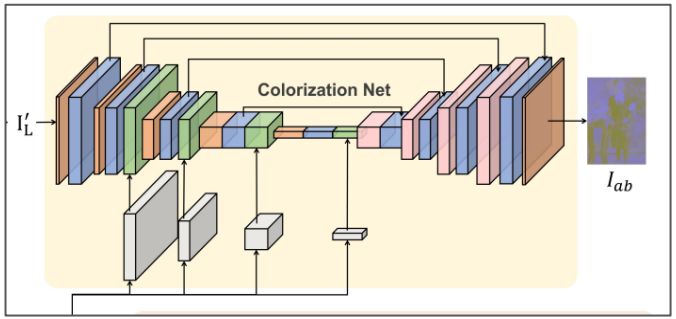
Warped SPHist

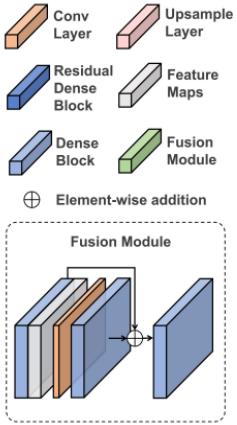
- Reshape into $\bar{h} \in \mathbb{R}^{HW \times k}$
- Downscale into 4 scales
- matrix multiplication with corresponding size of similarity map

Methodology

3. Colorization Sub-Net

2. Colorization Net





- U-net shape, densenet blocks
- Fusion module combines color heuristic and deep features
- Output I_{ab} is concatenated with I'_L to produce the final output $I_{Lab} \in \mathbb{R}^{H \times W \times 3}$

Methodology

Training Objective

Weighted sum of diverse objective functions

$$\mathcal{L} = \alpha\mathcal{L}_{rec,L} + \beta\mathcal{L}_{perc,L} + \lambda\mathcal{L}_{\mathrm{EMD},\hbar} + \gamma\mathcal{L}_{rec,ab} + \eta\mathcal{L}_{adv,ab}.$$

1. Luminance reconstruction loss $\mathcal{L}_{rec,L} = ||I_L^{'} - G_L||_1.$

2. Perceptual loss $\mathcal{L}_{perc,L} = \sum_j \frac{1}{C_j H_j W_j} ||\phi_j(I_L^{'}) - \phi_j(G_L)||_2^2,$

3. Histogram loss $\mathcal{L}_{\mathrm{EMD},\hbar} = \sum_{k=1}^{K} (\mathrm{CDF}_{\hbar_{I'}}(k) - \mathrm{CDF}_{\hbar_R}(k))^2,$

4. Chroma reconstruction loss $\mathcal{L}_{rec,ab} = ||I_{ab}^{'} - G_{ab}||_1.$

5. Adversarial loss $\mathcal{L}_{adv,ab} = \mathbb{E}_{G_{ab}}[\log D(G)] + \mathbb{E}_{I_{ab}^{'}}[\log(1 - D(I^{'}, R))].$

Experiments

Evaluation Metrics

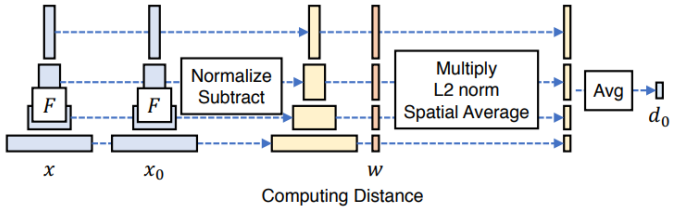
Goal: predict a quality score that correlates well with human perception

PSNR ↑ (Peak Signal-to-Noise Ratio)

- Higher PSNR value indicates a higher reconstruction quality
- Simple comparison – doesn't comply with the results of the human visual system

SSIM ↑ (Structural Similarity Index)

- Measure the similarity between two images
- Consider luminance, contrast, structure



LPIPS ↓ (Learned Perceptual Image Patch Similarity)

- Judge the perceptual similarity between two images
- computes the similarity between the activations of two image patches for some network
- Matches human perception well

Experiments

Setting

- Adam solver: $\beta_1 = 0.99, \beta_2 = 0.999$
- Initial learning rate$= 0.0001$, decay rate$= 0.99$
- Fixed loss balance weights: $\alpha = 1.0, \beta = 0.2, \lambda = 0.5, \gamma = 1.0, \eta = 0.2$
- Random crop 256×256 patches
- 20 epochs
- Single GTX 3090Ti GPU

Dataset	Train	Valid/Test	Usage
Div2K	800	100/100	Colorization
Pascal VOC	10,000	1000	Restoration, colorization
RealOld	X	200	Testing models trained on Pascal

Train and evaluate model on three datasets: Div2K, Pascal, and RealOld

Experiments

Results

Quantitative Comparison

Table 1: Quantitative comparison on the DIV2K and Pascal VOC validation datasets. Up-ward arrows indicate that a higher score denotes a good image quality. We highlight the best score for each measure.

Dataset	DIV2K (w/o degradation)			Pascal VOC (w/o degradation)			Pascal VOC (w/ degradation)		
Metric	PSNR↑	SSIM↑	LPIPS↓	PSNR↑	SSIM↑	LPIPS↓	PSNR↑	SSIM↑	LPIPS↓
Pix2pix	21.12	0.872	0.138	20.89	0.782	0.200	20.37	0.732	0.231
DeOldify	23.65	0.913	0.128	23.96	0.873	0.117	21.45	0.789	0.192
He *et al.*	23.53	0.918	0.125	23.85	0.925	0.114	-	-	-
InstColorization	22.45	0.914	0.131	23.95	0.932	0.111	-	-	-
Wan *et al.* -	-	-	-	-	-	-	18.01	0.598	0.421
Ours	**23.95**	**0.925**	**0.120**	**24.01**	**0.940**	**0.100**	**22.22**	**0.828**	**0.186**

Table 2: Quantitative comparisons of restoration/colorization performance of the compared models on the RealOld dataset.

Dataset	Real old photo		
Metric	PSNR↑	SSIM↑	LPIPS↓
Pix2pix	16.80	0.684	0.320
DeOldify	17.14	0.723	0.287
He *et al.*	16.72	0.707	0.314
InstColorization	16.86	0.715	0.312
Wan *et al.*	16.99	0.709	0.303
Ours	**17.20**	**0.758**	**0.258**

Experiments

Results

Qualitative Comparison

Figure 3: Visual comparisons against state-of-the-art colorization methods on DIV2K. It shows that with only 800 training images, our method is able to accomplish visually pleasant colorization and our result is significantly better than others.

Experiments

Results

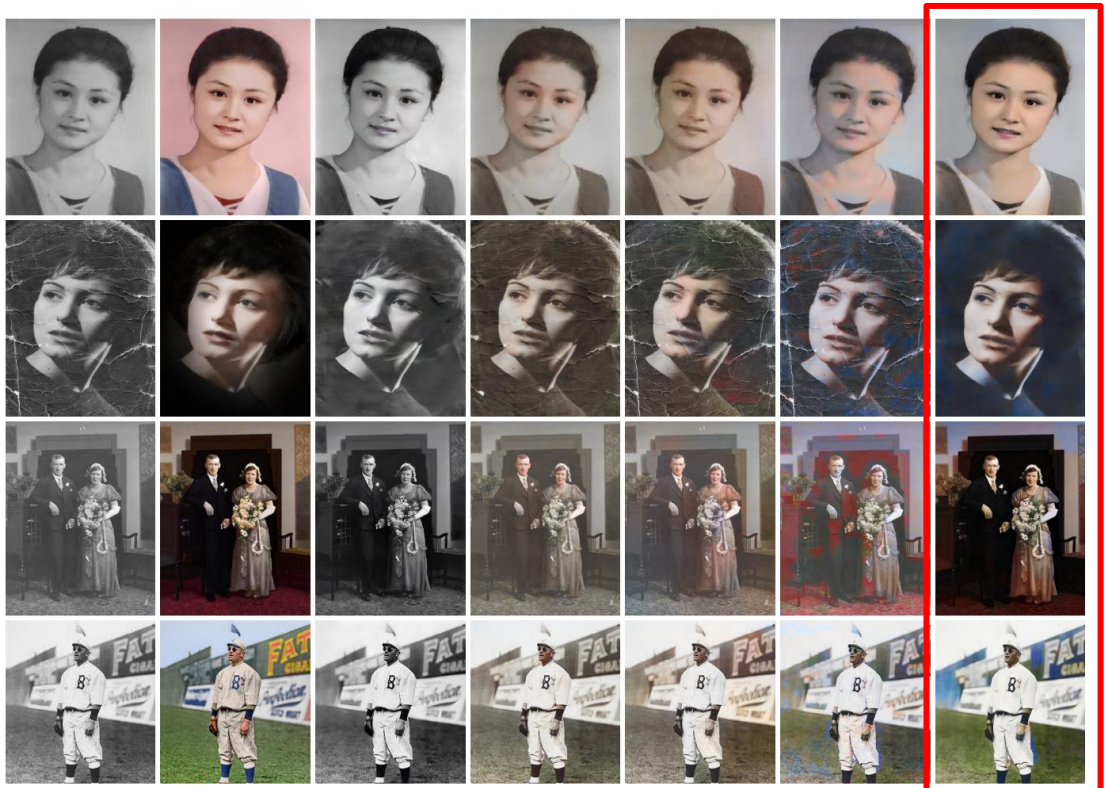


Figure 4: Visual comparisons against state-of-the-art colorization and restoration methods on RealOld dataset. It shows that with the limited synthetic training data from Pascal, our model is able to fix most of the degradation and deliver plausible colorization.

Experiments

Ablation Studies

Table 4: Ablation study of multi-scale SPHist on Div2k.

Method	PSNR↑	SSIM↑	LPIPS↓
Input *ab* fusion	22.978	0.902	0.130
Multi-scale *ab* fusion	23.233	0.910	0.127
Single-scale histogram fusion	23.631	0.906	0.125
Multi-scale histogram fusion	**23.952**	**0.925**	**0.120**

Table 5: Ablation study of multi-scale similarity maps on Div2k.

Method	PSNR↑	SSIM↑	LPIPS↓
No similarity map	22.817	0.910	0.131
Single-scale similarity map	23.803	0.922	0.126
Multi-scale similarity map	**23.952**	**0.925**	**0.120**

Table 6: Ablation study of multi-level RDN on Pascal witho degradation.

Method	PSNR↑	SSIM↑	LPIPS↓
Single-level RDN	21.89	0.818	0.190
Multi-level RDN	**22.22**	**0.828**	**0.186**

Thank You!