IMAGE COLORIZATION PROBLEM USING (VARIATIONAL) AUTO-ENCODER

Prepared by:

Khanh Duong, Tien-Huy Nguyen, Nhu-Tai Do

taidn@ueh.edu.vn

Agenda

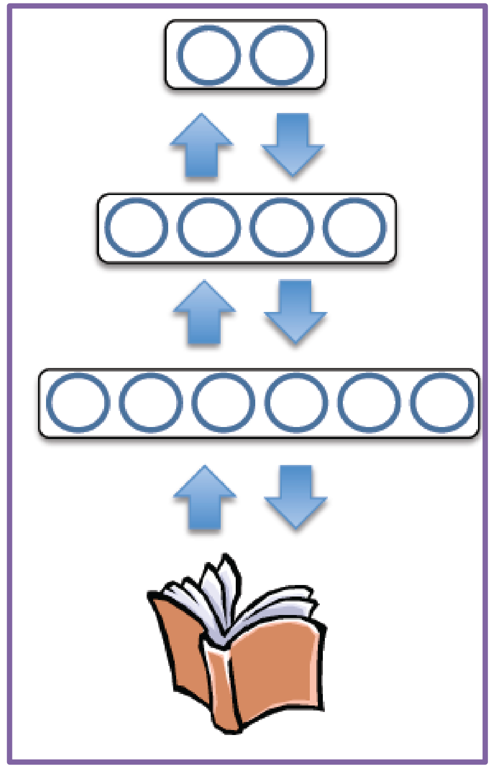
1. Introduction
2. Image Colorization Problem
3. Context Auto-Encoder Approach
4. Variational Auto-Encoder Approach

Introduction about (Variational) Auto-Encoder

Introduction

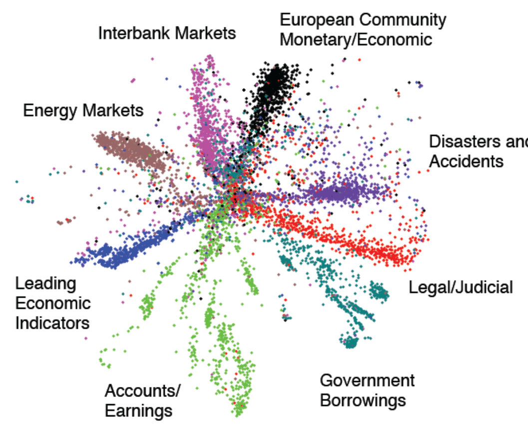
Massive increase in the amount of the data → Deep Unsupervised Model Learned latent code → Inference and discover structure at multiple levels

Mostly Unlabeled



Bag of Word

Reuters dataset: 804,414 newswire stories: **unsupervised**



(Hinton & Salakhutdinov, Science 2006)

underlying structure, cause, or statistical correlation

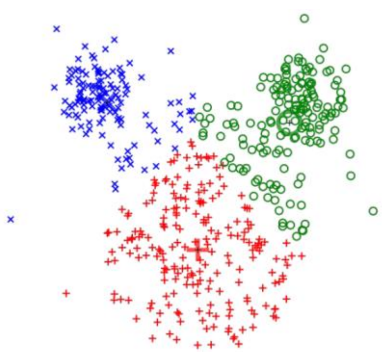
[1] Russ Salakhutdinov, Deep Unsupervised Learning, Slides, CMU

What is Unsupervised Learning?

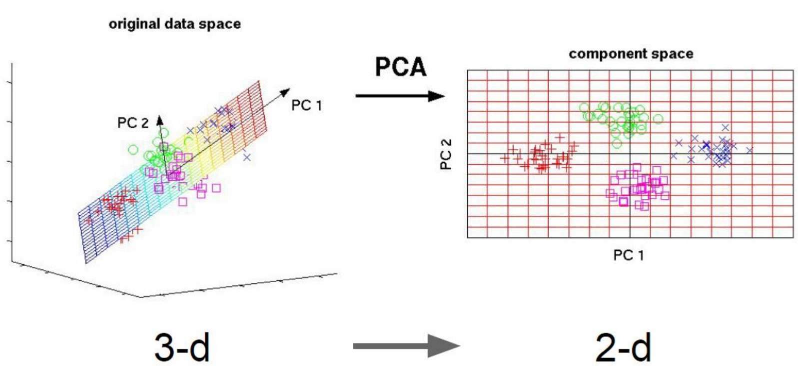
Solve unsupervised learning => understand structure of visual world

Training data is cheap

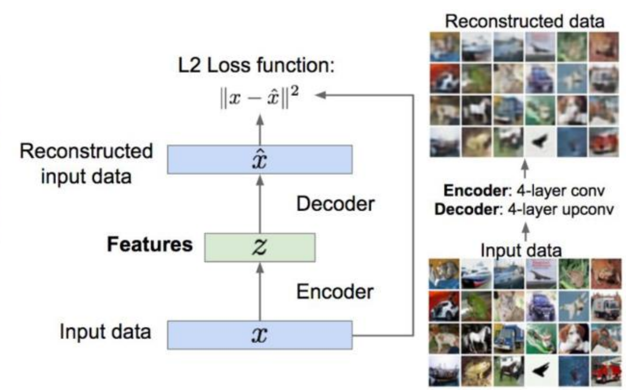
	Supervised Learning	Unsupervised Learning
Data	(x, y) - x is data, y is label	x - Just data, **no labels**
Goal	Learn a ***function*** to map x -> y	Learn some underlying ***hidden structure*** of the data
Application	Classification, Regression, Object detection, Semantic segmentation, Image Captioning, …	Clustering, Dimensionality reduction, Feature learning, Density estimation, …



Clustering



Dimensionality reduction



Feature learning

[2] *CS231n Convolutional Neural Networks for Visual Recognition, Standford*

Technical mind-map in Unsupervised Learning

- Unsupervised Learning
  - Non-probabilistic (Parametric) Models
    - Sparse Coding
    - **Autoencoders**
    - Others (e.g. k-means)
  - Probabilistic (Generative) Models
    - Explicit Density p(x)
      - Tractable Density
        - Fully observed Belief Nets
        - NADE
        - PixelRNN
      - Approximate Density
        - Boltzmann Machines
        - **Variational Autoencoders**
        - Helmholtz Machines
        - Many others…
    - Implicit Density p(x)
      - Generative Adversarial Networks
      - Moment Matching Networks
      - Markov Chain

Figure copyright and adapted from Ian Goodfellow, Tutorial on Generative Adversarial Networks, 2017

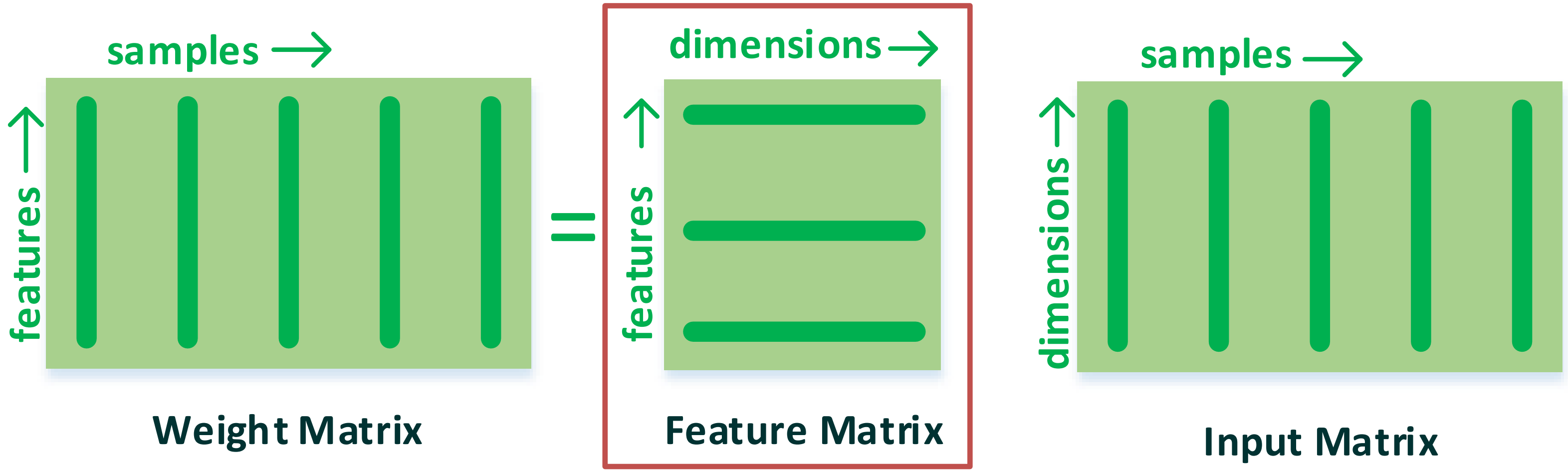
[1] Russ Salakhutdinov, Deep Unsupervised Learning, Slides, CMU

Dimensionality Reduction Problem

- Given input data **X** with **N** samples in **D** dimension space

$$X = X_N = \{x_1, x_2, \dots, x_N\}, x_i \in \mathbb{R}^D$$

- Find feature matrix **W**: $W = W_M = \{w_1, w_2, \dots, w_M\}, w_i \in \mathbb{R}^D$
- Use **W** to transform **X** into weight matrix $\tilde{Z}$: $\tilde{Z} = W^T X$



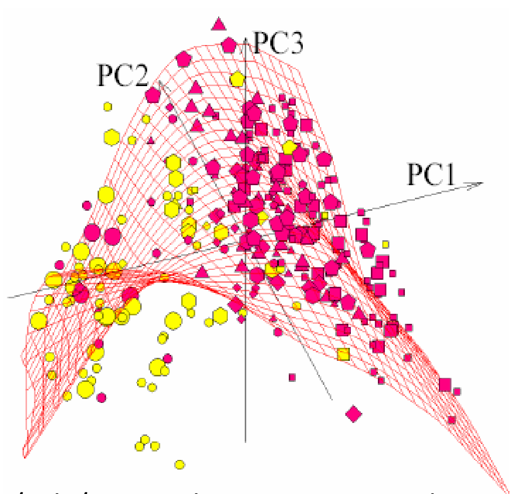
- Find a good representation?
- Reduce redundancy in the data?

Dimensionality Reduction Problem

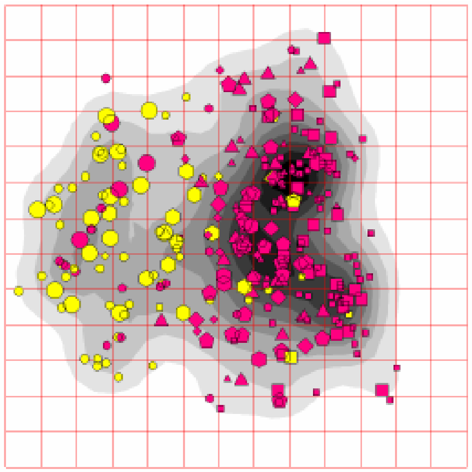
- Desirable feature features:
  - Avoid feature similarity → $w_i^T w_j = 0$ → linear combination
  - Give "simple" weights → $Cov(z_i, z_j) = I$ → minimize relation of the two dimensions
- Satisfy minimising the total squared reconstruction error:

$$\|W_D X - W_M X\|_2 \rightarrow min$$

Where $M \ll D, W_M \subset W_D$



PCA 2D



[3] https://en.wikipedia.org/wiki/Principal_component_analysis

Feature Learning

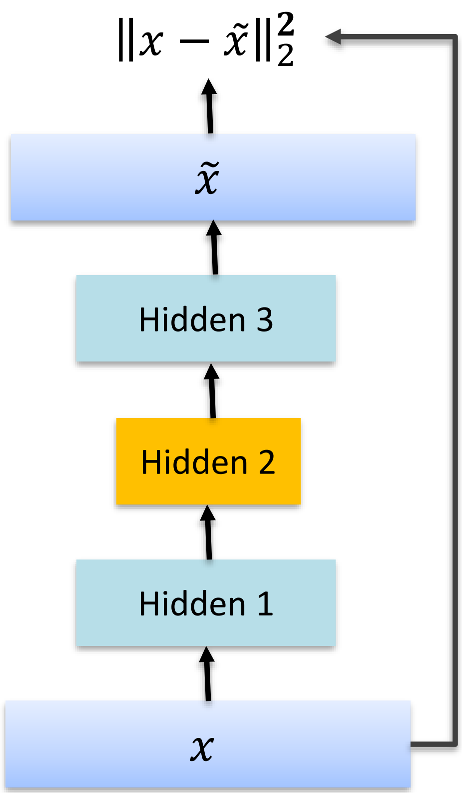
Motivation

- Training very deep neural networks is difficult:
  - ***Magnitudes of gradients*** in ***lower*** layers and in ***higher layers*** are ***different***
  - The landscape of objective function is ***difficult*** for SGD to ***find a good local optimum***
  - Many parameters to remember training data and do ***not generalize well***
- The goal of pretraining is to address the above problems:
  - ***Pretraining step***: train a sequence of shallow autoencoders, greedily one layer at a time, using unsupervised data
  - ***Fine-tuning step 1***: train the last layer using supervised data
  - ***Fine-tuning step 2***: use backpropagation to fine-tune the entire network using supervised data

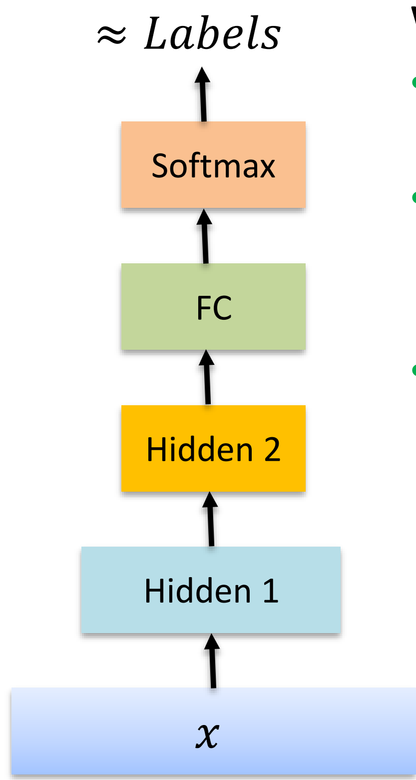
[7] Quoc V.Le, ***A Tutorial on Deep Learning Part 2: Autoencoders, Convolutional Neural Networks and Recurrent Neural Networks***, Google Brain, robotics.stanford.edu/~quocle/tutorial2.pdf

Feature Learning

General Architecture



Phase 1
Train the Autoencoder
using all data



Phase 2
Train the Autoencoder
on “labeled” data

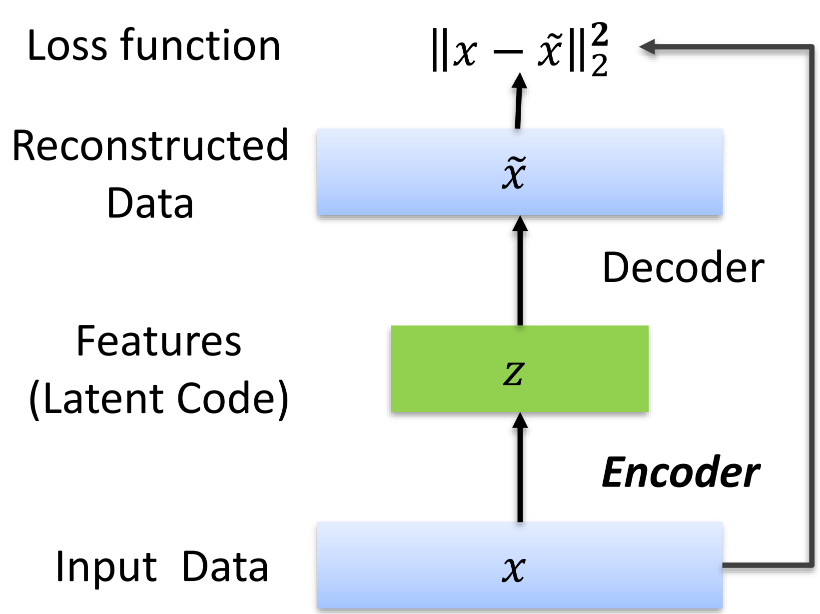
Work often better:

- ***learns internal data representation***: may be useful features
- ***initializes optimization*** from more favorable initial approximation: good for solving vanishing gradient problem
- ***especially useful when few labelled examples*** and many unlabeled

[8] T.Paine, ***An analysis of unsupervised pre-training in light of recent advances***, ICLR 2015

General AutoEncoders

- Autoencoders: artificial neural networks
  - Capable of learning efficient representations of the input data, called ***latent code***
  - Without any supervision, simply learning to ***reconstruct original data***
  - ***Need to constrain complexity***: (1) by ***architectural constraint*** (2) by penalty on ***internal representation***



Goal: Train such that features used to reconstruct original data, don't use labels

Hidden layer z: features

+ smaller than x (dimensionality reduction)
+ sparse constraint (larger than x)

Encoder, Decoder:

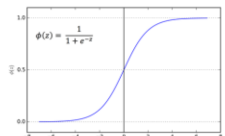
+ Linear + Nonlinearity (sigmoid)
+ Deep, fully – connected
+ ReLU CNN

[2] CS231n Convolutional Neural Networks for Visual Recognition, Standford

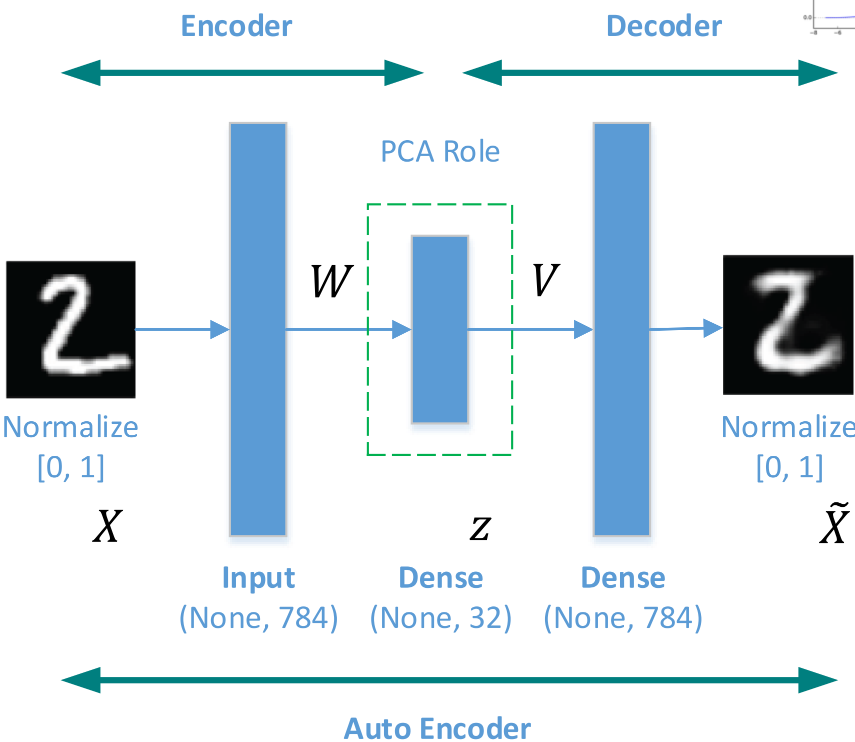
Vanilla (Undercomplete) AutoEncoder

$$Loss(I, \hat{I}) = -\sum (\mathrm{I}(x) - \hat{\mathrm{I}}(x))^2$$

$$Loss(I, \hat{I}) = -\sum \mathrm{I}(x) \log(\hat{\mathrm{I}}(x)) + (1 - \hat{\mathrm{I}}(x)) \log(\mathrm{I}(x))$$



Sigmoid



+ Encoding: X (input data), f (activation function)

$$z = \boldsymbol{f}(WX)$$

+ Decoding: g (activation function)

$$\tilde{X} = \boldsymbol{g}(Vz) = \boldsymbol{g}(V\boldsymbol{f}(WX))$$

+ If **g**, **f** is linear function:

$$\tilde{X} = VWX$$

+ Loss function MSE:

$$\min_{W,V} \|X - \tilde{X}\|$$

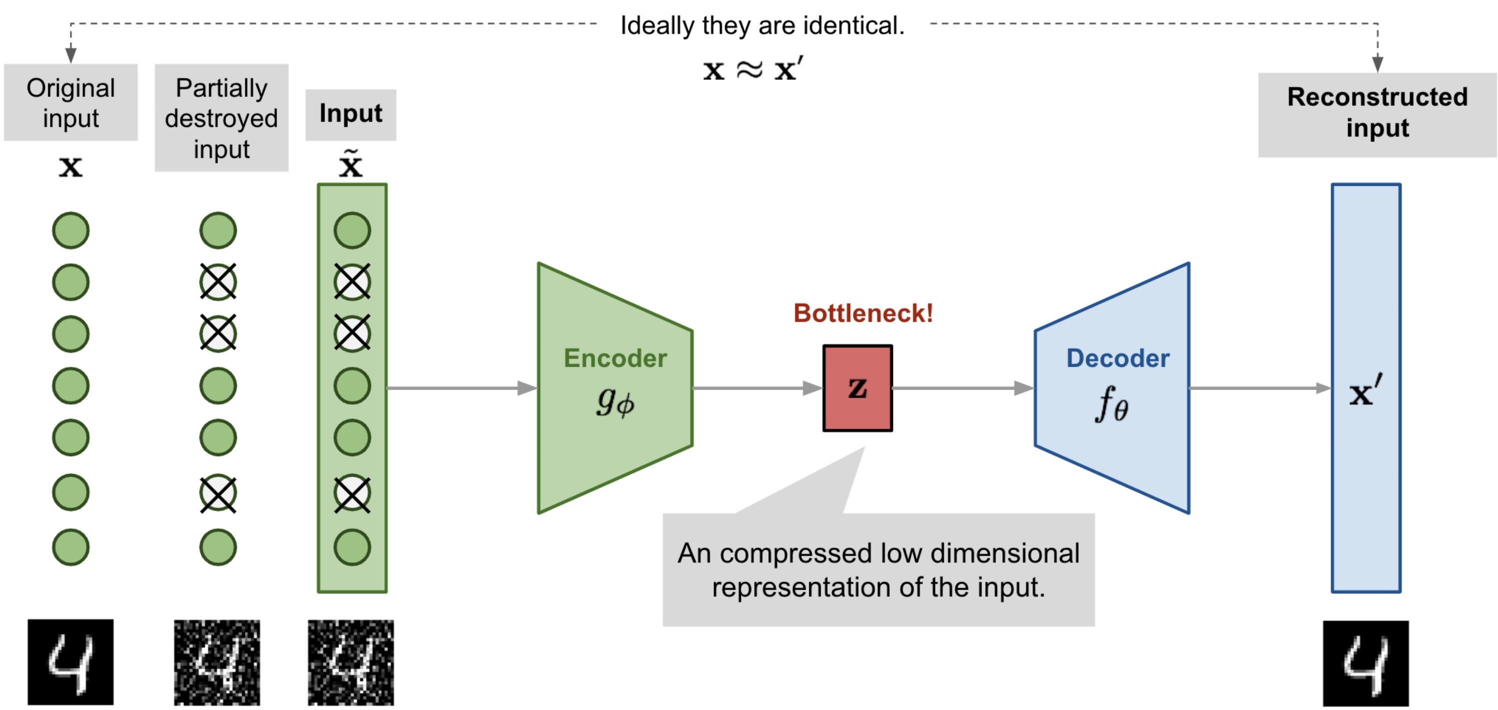
$$\min_{W,V} \|X - VWX\|$$

Dimensionality reduction with z as new subspace for input data X, ability reconstruct X with $\tilde{X}$.

If ***g***, ***f*** is non-linear function (sigmoid) → Non-Linear PCA

Denoising Autoencoder

- To avoid overfitting and improve the robustness, the input is partially corrupted by adding noises to or masking some values of the input vector in a stochastic manner



Variational Autoencoder

- Instead of mapping the input into a *fixed* vector, we want to map it into a distribution.

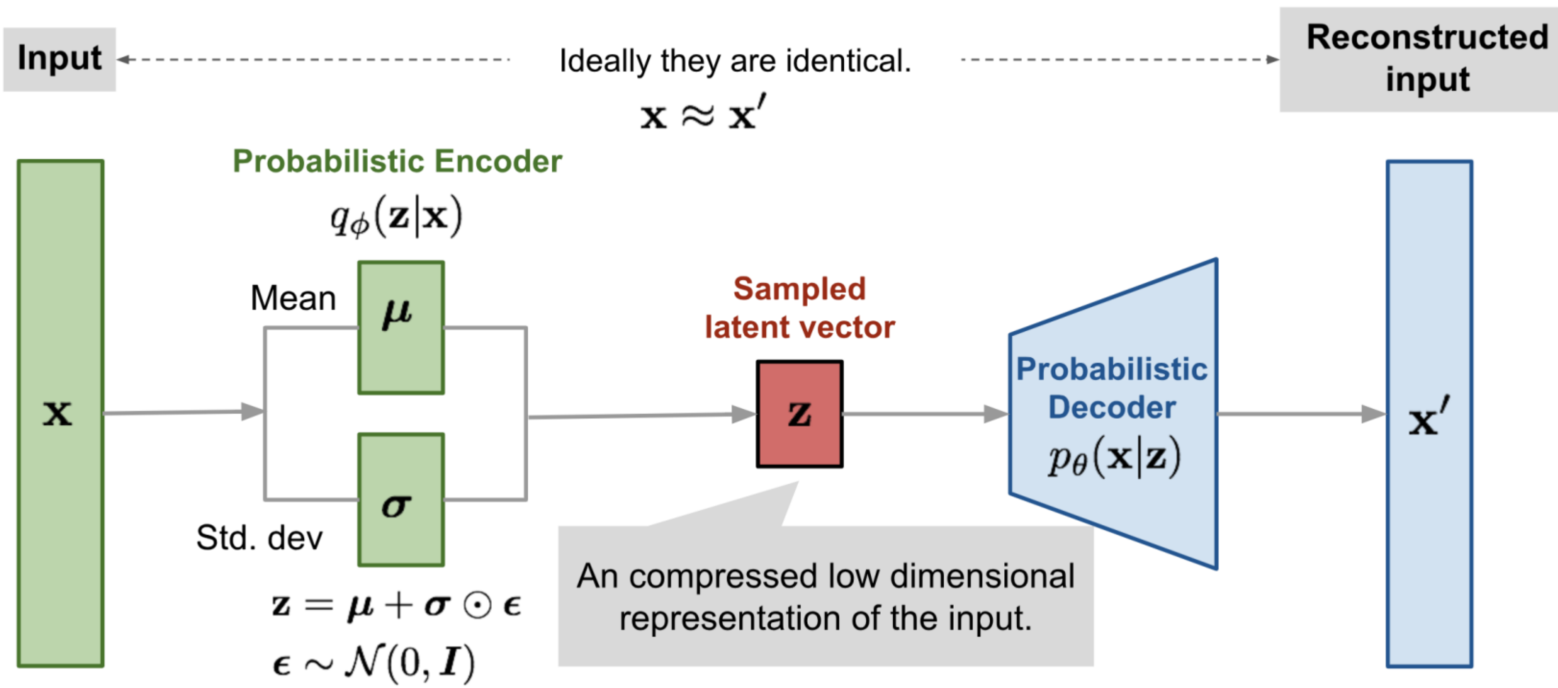
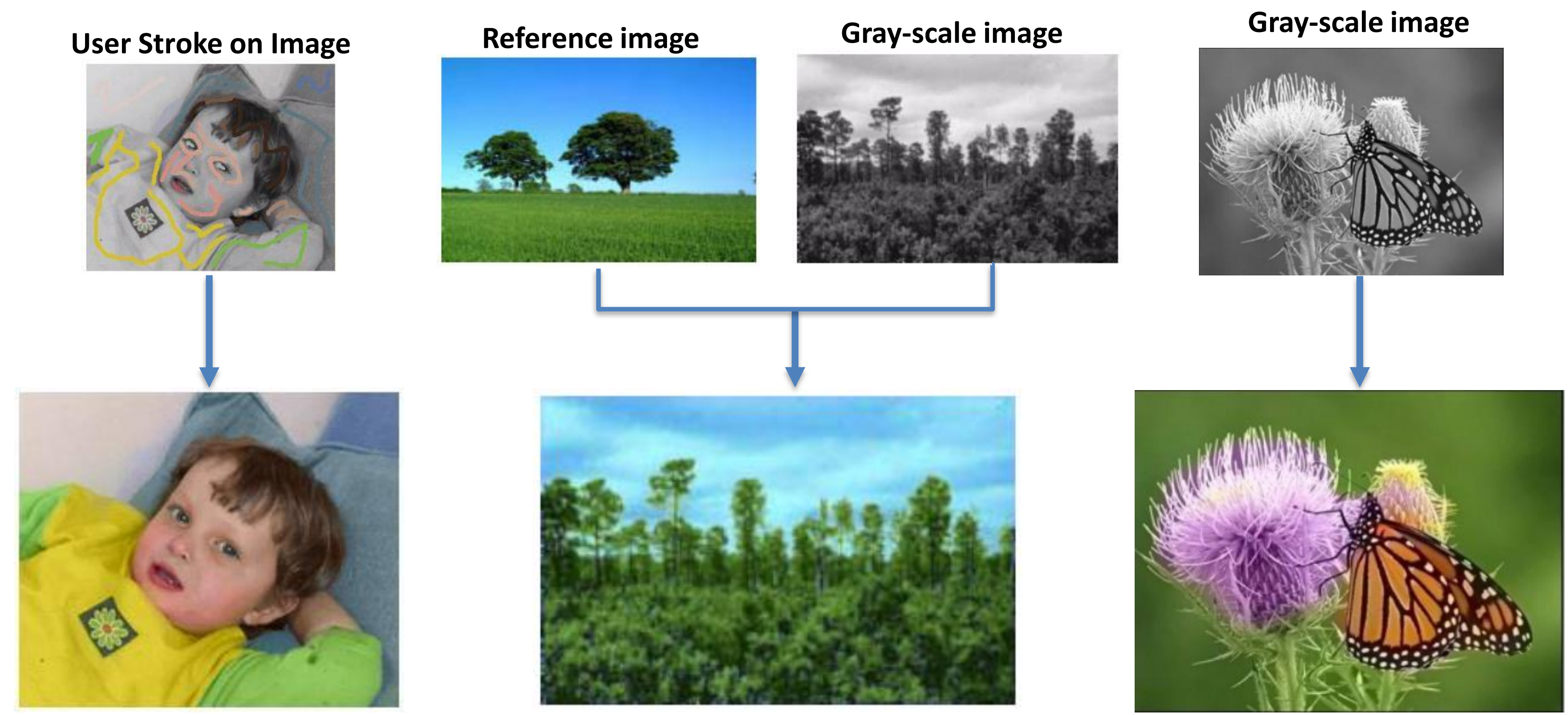


Image Colorization Problem

Introduction

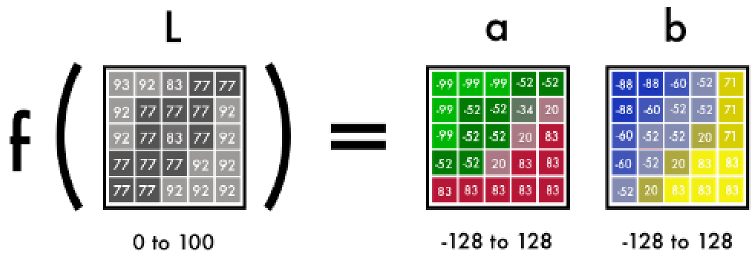
- **Problem**: Image Colorization is the task of colorizing gray-scale images.
- **Practical applications**: coloring old black and white images, movies etc.
- **Main approaches:** Scribble-based, Example-based, and **<u>Fully Automatic</u>**.



Scribble-based colorization | Example-based colorization | Fully Automatic colorization

Introduction

- **Our problem** focuses on **Fully Automatic Colorization**: Given the **grayscale image**, produce ***a plausible colorization to fool a human observer***.
  - **Input**: Grayscale image in grids of pixels from 0 – 255
  - **Output**: Channel a, b of color image in CIE Lab color space

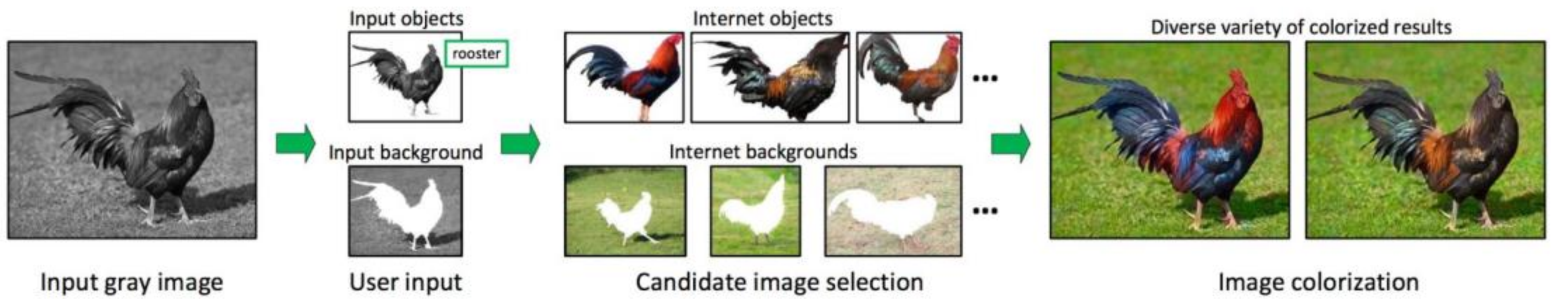


  - 94% of the cells in our eyes determine brightness, only 6% for colors → grayscale image is a lot sharper than the color layers.

https://www.freecodecamp.org/news/colorize-b-w-photos-with-a-100-line-neural-network-53d9b4449f8d/

Related works

- **Non-parametric methods**: transfer color reference images onto input image from analogous regions

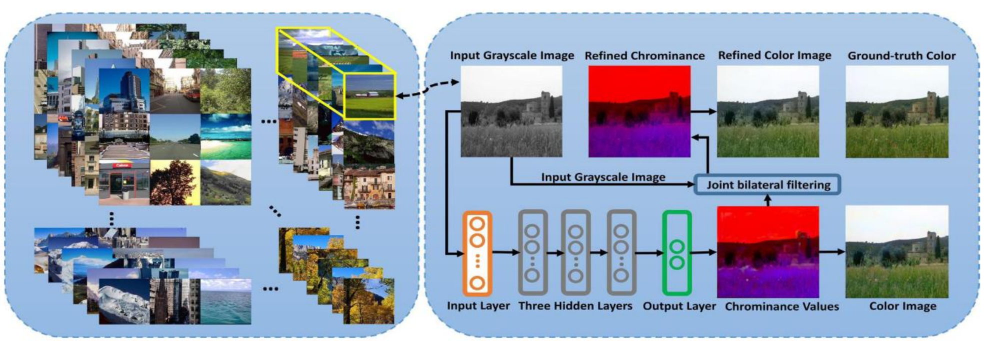


- **Parametric methods**: learn prediction functions from large datasets
  - **Problem define**: (1) **regression** onto continuous color space, (2) **classification** of quantized color values
  - **Approach**: (1) Hand-engineered Features (2) Deep networks

Related works

- **Parametric methods:** Hand-engineered Features
  - **Cheng et al.[1]**: ***adaptive image clustering*** according to global information, every ***neural network*** trained on specific cluster for colorization with ***L2 Regression loss***, using joint bilateral filtering for post-processing.
  - **Charpiat et al.[2]**: ***deal with multimodality*** in colorization with the probability distribution of all possible colors on every pixel, ***use graph-cut*** to maximize the probability, ***discretization of the color space***.

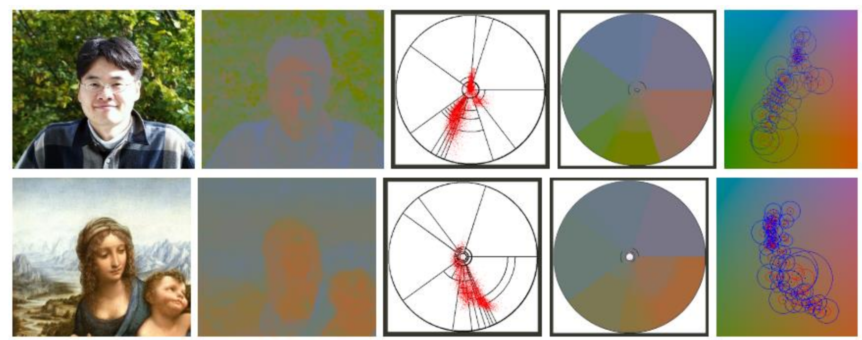
Regression



(a) Reference image clusters (b) The proposed colorization method

Cheng et al.

Classification



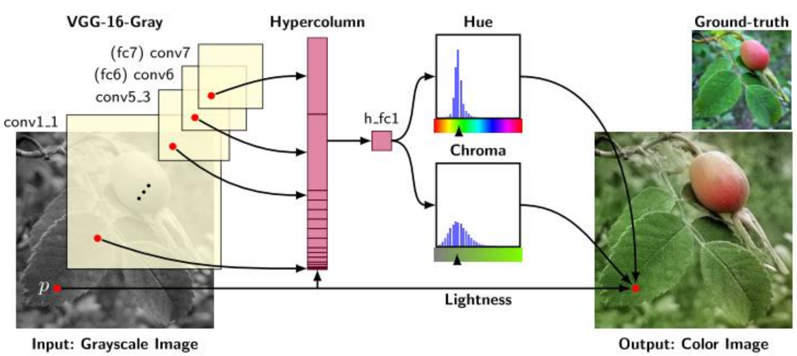
Charpiat et al.

[1] Z. Cheng, Q. Yang, and B. Sheng, "***Deep colorization***," *IEEE International Conference on Computer Vision*, pp. 415–423, 2015.
[2] G. Charpiat, M. Hofmann, and B. Schölkopf, "Automatic image colorization via multimodal predictions," *Lecture Notes in Computer Science (including subseries Lecture Notes in Artificial Intelligence and Lecture Notes in Bioinformatics)*, vol. 5304 LNCS, pp. 126–139, 2008.

Related works

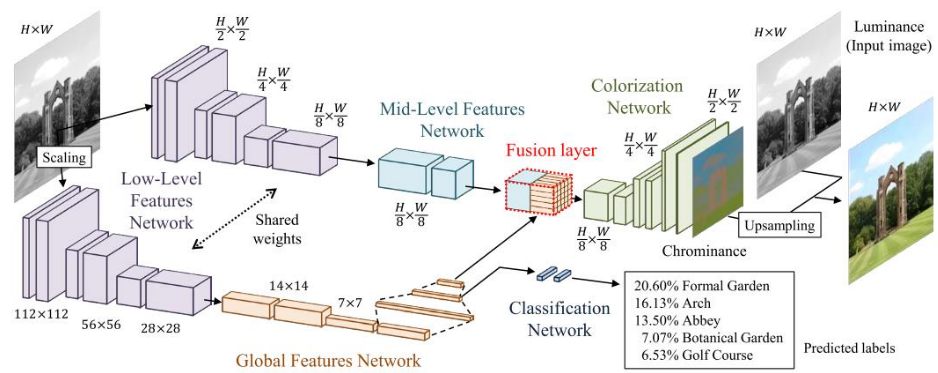
- **Parametric methods:** Deep Learning Approach:
  - **Larsson et al.**[1]: use un-rebalanced classification **loss**, build on hypercolumns on a VGG **network**, train on **ImageNet**, evaluate on **PSNR, RMSE**.
  - **Iizuka et al.**[2]: use a regression **loss**, build a ***two-stream architecture*** fusing global and local features, train on ***Places scene dataset***, evaluate on ***naturalness*** of the colorizations by ***user asking***

Regression



Larsson et al.

Classification



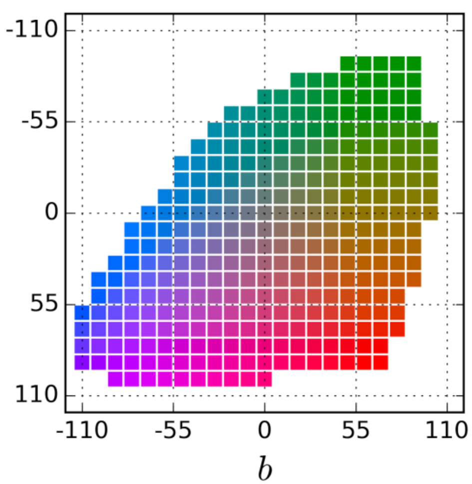
Iizuka et al.

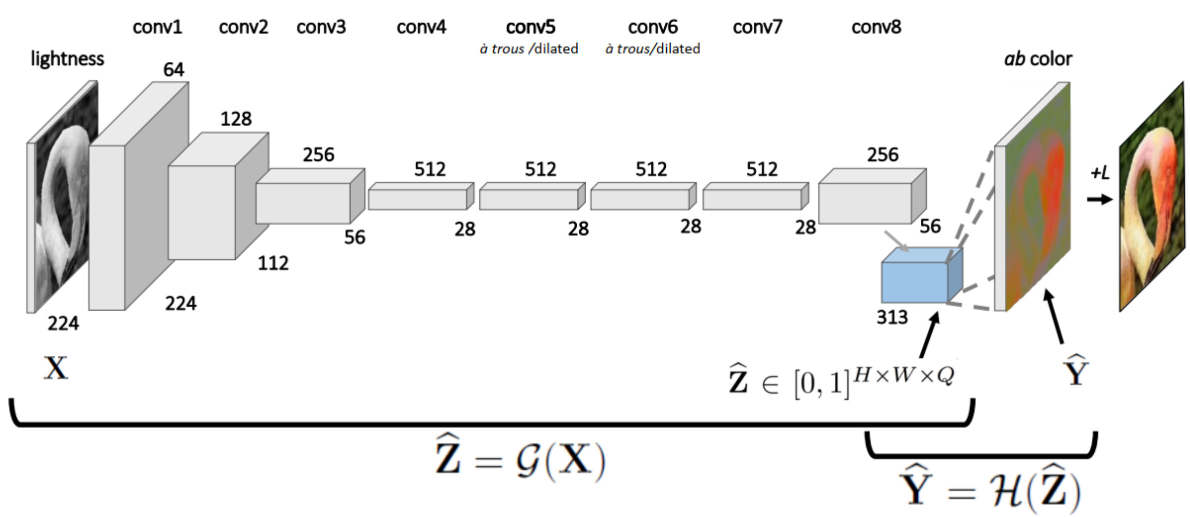
[1] G. Larsson, M. Maire, and G. Shakhnarovich, "***Learning Representations for Automatic Colorization***," in *Lecture Notes in Computer Science (including subseries Lecture Notes in Artificial Intelligence and Lecture Notes in Bioinformatics)*, vol. 9908 LNCS, 2016, pp. 577–593.
[2] S. Iizuka, E. Simo-Serra, and H. Ishikawa, "***Let there be Color: Joint End-to-end Learning of Global and Local Image Priors for Automatic Image Colorization with Simultaneous Classificatio***," *ACM Transactions on Graphics*, vol. 35, no. 4, pp. 1–11, Jul. 2016.

Related works

- **Zhang et. at[1]: Main idea**
  - ***Multinomial classification*** problem by ***quantize ab space*** into grid size 10, keep 313 bins in gamut.
  - Cross entropy loss with ***class rebalancing*** to encourage learning of rare colors.
  - **Post-processing**: per-pixel color distribution to single point estimate **by interpolating between mean and mode with annealed-mean.**

Colors in *ab* space
(discrete)





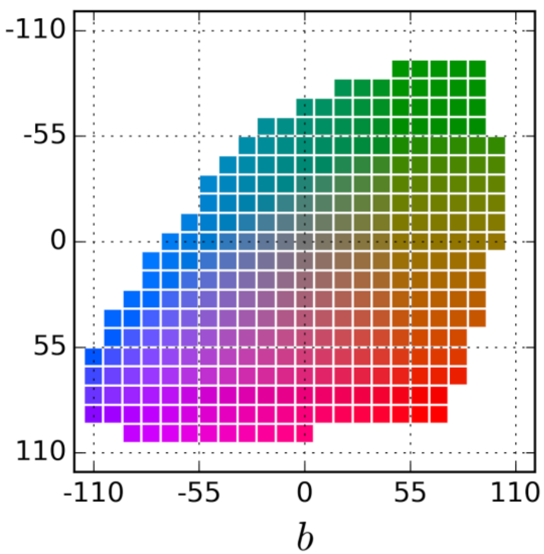
Deep network model

[1] R. Zhang, P. Isola, and A. A. Efros,**Colorful Image Colorization**, ECCV, pp. 649–666, 2016

Related works

- **Quantization process** in classification approach from Richard Zhang et. al.:
  - ***Quantization*** Lab Color Space ***into 313 bins***
  - Using ***soft-encoding scheme*** instead of nearest searching
- Benefits from this quantization process to classify:
  - Prevent the averaging effect of regression loss: easy to favor grayish, desaturated results
  - Increase the correlation between nearest color pixels by soft-encoding.

Colors in *ab* space
(discrete)



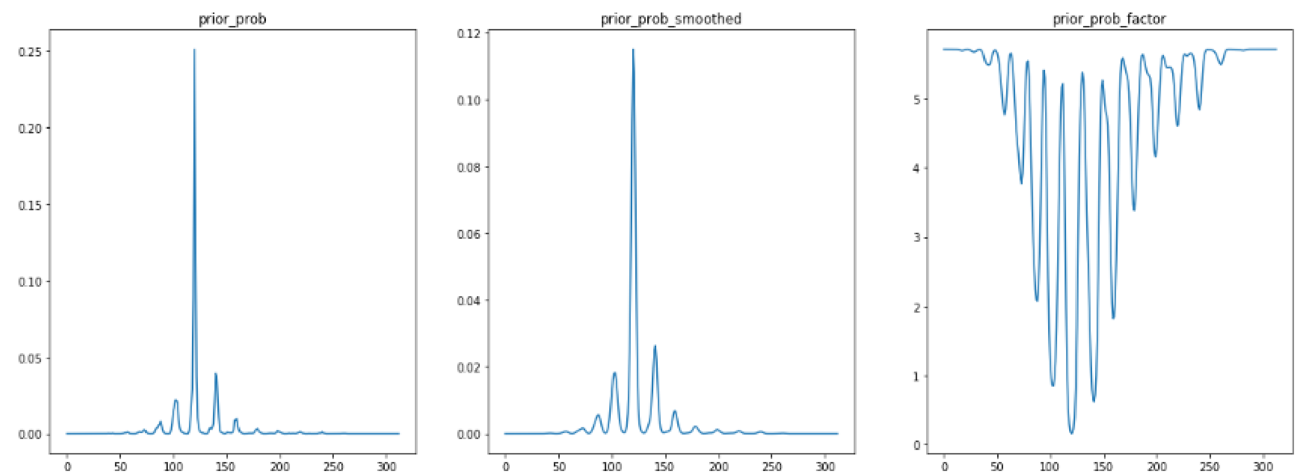
$$L(\hat{Z}, Z) = -\frac{1}{HW}\sum_{h,w} v(Z_{h,w}) \boxed{\sum_{q} Z_{h,w,q}} log(\hat{Z}_{h,w,q})$$

Rarity weighting　Target distribution　Predicted distribution

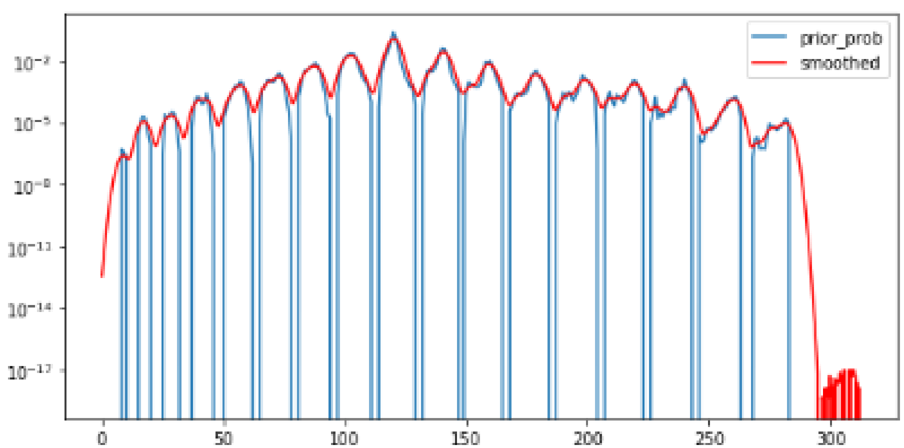
Category Cross entropy loss

Related works

- **Smoothing the color prior probability**:



Smoothness of color probability, Invert Probability

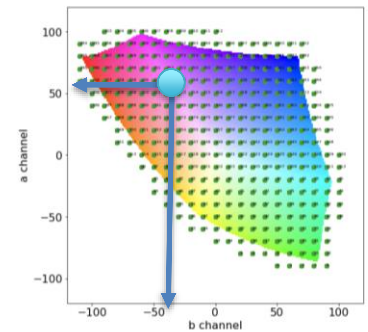


Distribution of probability vs smoothness probability

Related works

- **More details: The ab color distribution**
  - **Soft-Encoding Process:**
    - **Step 1**: For every pixel of image, convert from ab values to color index q (encoding) using K-Nearest

$$I_{ab}(p) = (a, b) \longrightarrow q \in [0,312]$$



quantize ab space with grid size 10 (313 bins)

- **Step 2**: Convert to one-hot encoding representation

0 … 312

	…	0	0	1	0	0	…	

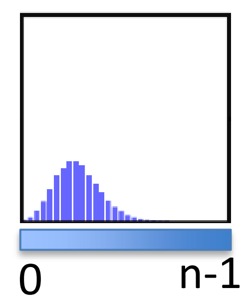
q

- **Step 3**: Apply label smoothing
  - Use K-Nearest neighbors to get 4 color indexes nearest q,
  - Generate 5 gaussian values, and normalize

0 … 312

	…	0.05	0.24	0.42	0.24	0.05	…	

q



Context Auto-Encoder Approach

Challenges

- **Averaging effect**: grayish, desaturated results due to 94% of the cells in our eyes determine brightness, only 6% for colors. Grayscale image is a lot sharper than the color layers.

- **Rare colors in images**: strongly biased due to the appearance of backgrounds such as clouds, pavement, dirt, and walls.

- **Semantic information matters**: In order to colorize any kind of image, a system must interpret the semantic composition of the scene (what is in the image: faces, cars, plants, . . .) as well as localize objects (where things are).

GT: lagoon
top-1: balcony interior (0.136)
top-2: beach house (0.134)
top-3: boardwalk (0.123)
top-4: roof garden (0.103)
top-5: restaurant patio (0.068)

Context-Aware Colorization

- **Objectives**:
  - Integrate scene-context classification and pixel-wise semantic segmentation

Grayscale Image

Color Image

scene-context classification
+ global scene information

Scene-Context Classification
(Label Id, Probability, Label Name)
310 - 0.49932244 - soccer_field
254 - 0.15201965 - park
164 - 0.12514195 - golf_course

Scene-context classes
(totally 365 classes)

Label Mask

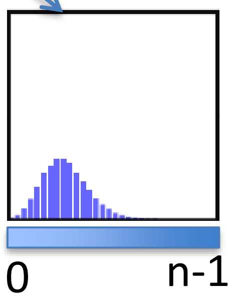
pixel-wise semantic segmentation
+ what object the pixels belong to

Segmentation classes in Coco-Stuff
(0: unlabeld, 1 – 182: objects & stuffes)

Context-Aware Colorization

- **Objectives**:
  - Use ab color distribution ***to encourage rare color (rebalancing colors), and multi-modal in colorization***

With a pixel

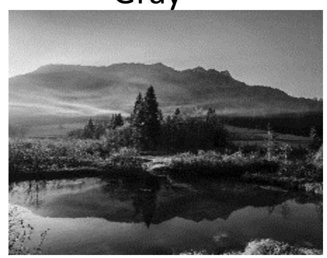


ab color distribution
vs.
ab color value

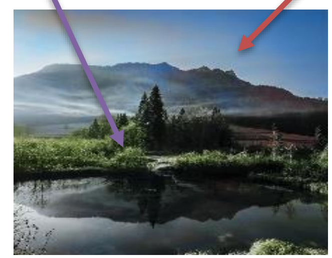
Tree

Sky (common colors)

Gray

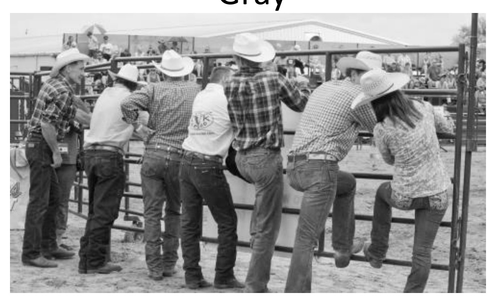


Ours



GT

Gray



GT

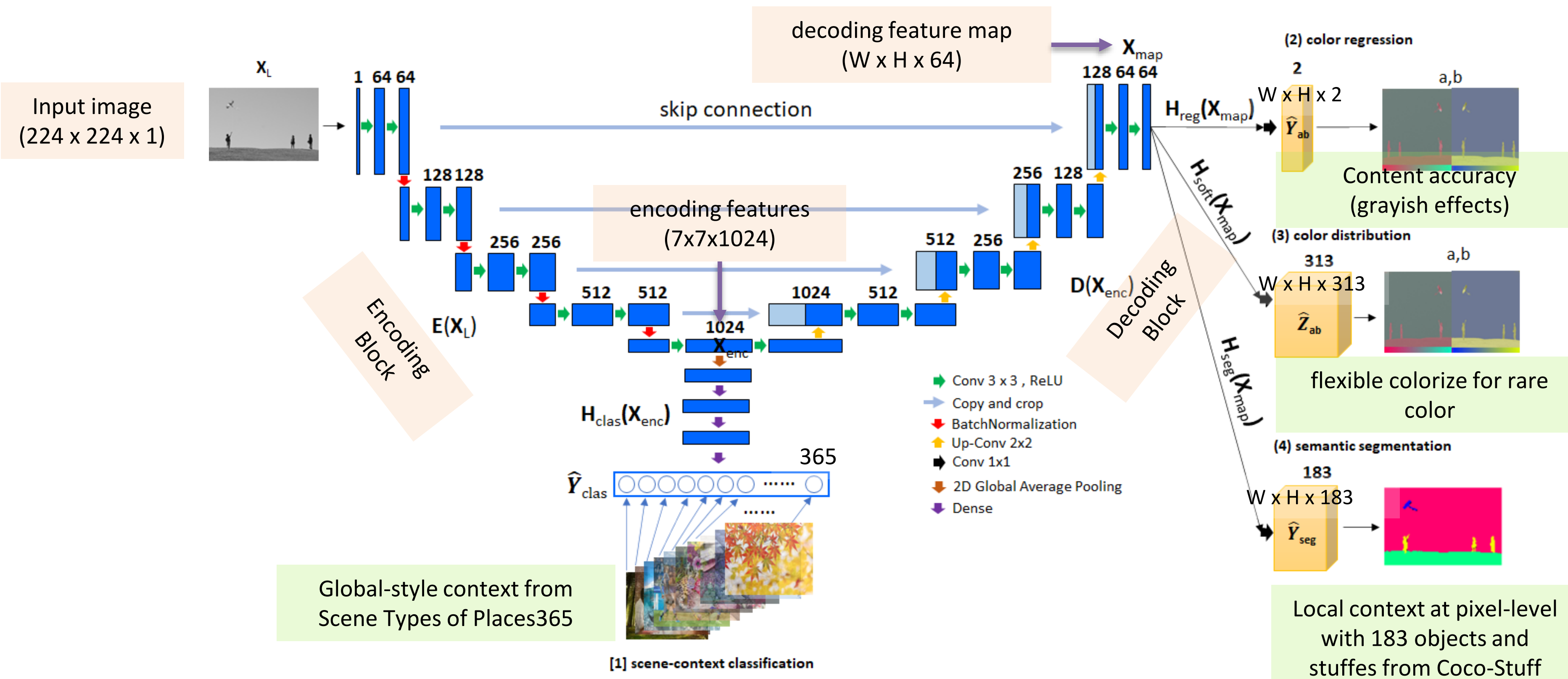
Shirt (diversity colors, rare colors)

Grayish result

Multi-Modal Attribute or Bias
(many choice in colorization)
leading to
Grayish or Desaturated Effect

Semantic Image Colorization Auto-Encoder

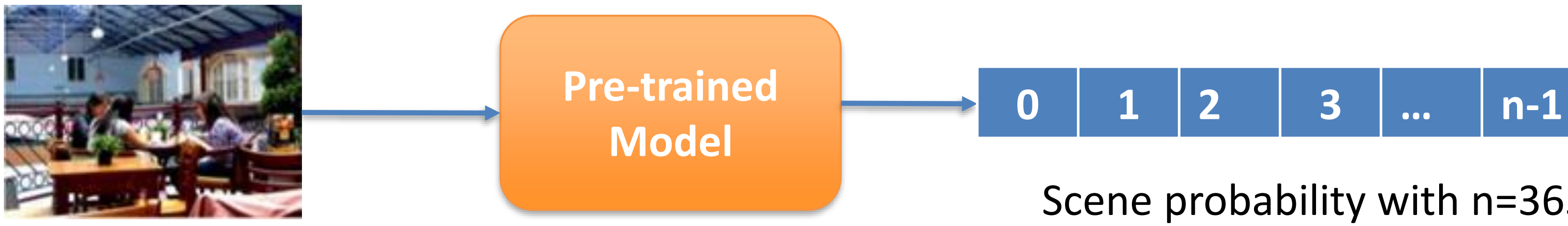
- **Take advantage of skip connections** between the contracting and expanding path at the same depth level using U-Net model (prevent dying ReLU and vanishing problem)
- **Use multi-task learning** with **end-end training** from **gray-scale image** to **four outputs** for learning mutual benefits of global/local context, content accuracy and color biases.



[1] Nhu-Tai Do et al. "***Image colorization using the global scene-context style and pixel-wise semantic segmentation***." IEEE Access 8 (2020): 214098-214114.

Scene-context classification

- Extract the scene probabilities of training dataset (without scene-context ground-truth) based on **pre-trained model on Places365**[1].



Image

Scene probability with n=365

- **Label Smoothing**[2] **with top-5 prediction**: keep 5 highest probabilities, set all remain values to 0, and normalize the probabilities with sum 1.

Top-1: Cafeteria (0.179)
Top-2: Restaurant (0.167)
Top-3: dining_hall (0.091)
Top-4: coffe_shop (0.086)
Top-5: restaurant_patio (0.080)

Top-1: Cafeteria (0.297)
Top-2: Restaurant (0.277)
Top-3: dining_hall (0.151)
Top-4: coffe_shop (0.143)
Top-5: restaurant_patio (0.132)

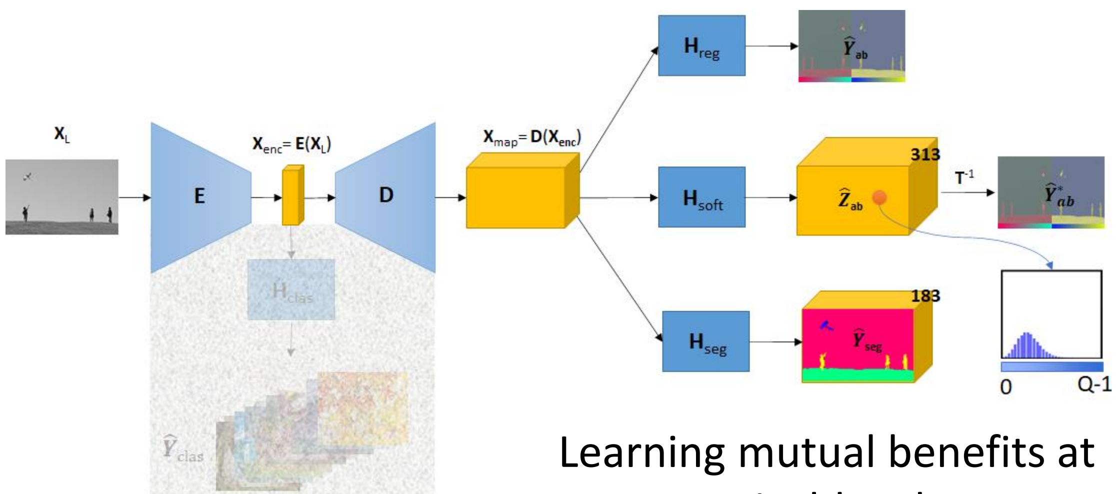
[1] B. Zhou, A. Lapedriza, A. Khosla, A. Oliva, and A. Torralba, “Places: A 10 Million Image Database for Scene Recognition,” IEEE transactions on pattern analysis and machine intelligence (TPAMI), vol. 40, no. 6, pp. 1452–1464, 2018
[2] R. Müller, S. Kornblith, and G. Hinton, “***When Does Label Smoothing Help?***,” In Advances in Neural Information Processing Systems (NeurIPS), pp.4696-4705, 2019.

Regression/Color Distribution/Segmentation Branches

- Compute **backward gradients** of **three branches** to enhance decoding feature map $\mathbf{X}_{map}$ and encoding feature $\mathbf{X}_{enc}$
  - **regression branch** to keep the accuracy between prediction/ground-truth → **output results** with grayish and desaturated effects (not used as colorized result)
  - **color distribution branch** to encourage rare color (rebalancing colors) and multi-modal in colorization → output results with more vivid
  - **segmentation branch** to help the system understand what object the pixels belong to (with 183 object & stuff labels) → output results with more precise edge

Gray

Ground-Truth

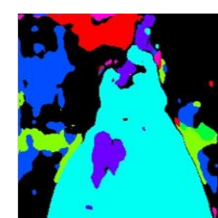


Learning mutual benefits at pixel-level

Reg

Soft

colorize result



Seg

Quantitative comparisons

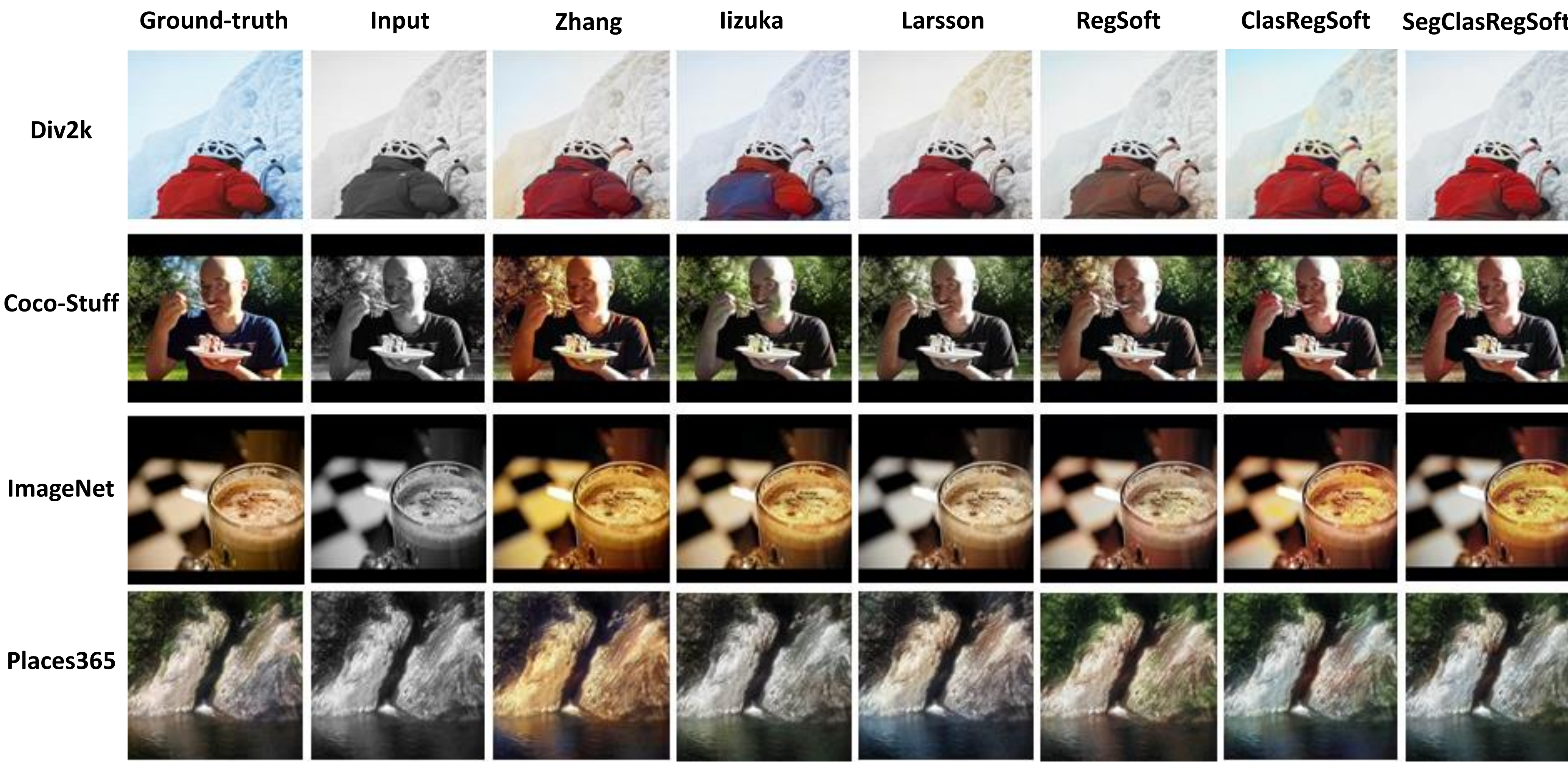
Method	ImageNet ctest1k			DIVK2K		
	PSNR ↑	SSIM ↑	$L2_{ab}$ ↓	PSNR ↑	SSIM ↑	$L2_{ab}$ ↓
Iizuka et al. [7]	22.841	0.865	0.277	22.981	0.919	0.079
Larsson et al. [8]	**23.335**	**0.869**	**0.26**	**23.490**	**0.929**	0.072
Zhang et al. [11]	21.297	0.848	0.286	20.929	0.896	0.079
Ours with RegSoft	22.102	0.896	0.269	22.026	0.914	0.071
Ours with ClassRegSoft	21.068	0.886	0.274	21.694	0.912	0.071
Ours with SegClassRegSoft	21.900	0.893	0.264	22.330	0.917	**0.068**

Method	Place365 ctest1k			COCO-Stuff ctest1k		
	PSNR ↑	SSIM ↑	$L2_{ab}$ ↓	PSNR ↑	SSIM ↑	$L2_{ab}$ ↓
Iizuka et al. [7]	**25.572**	**0.948**	0.481	23.541	0.871	0.242
Larsson et al. [8]	25.096	0.945	0.452	**23.773**	0.873	**0.223**
Zhang et al. [11]	23.076	0.928	0.484	21.502	0.851	0.245
Ours with RegSoft	23.599	0.932	0.474	22.872	0.912	0.23
Ours with ClassRegSoft	22.916	0.924	0.466	22.134	0.907	0.23
Ours with SegClassRegSoft	23.858	0.931	**0.442**	22.985	**0.913**	**0.223**

- Larsson et al.: better on PSNR for ImageNet,DIV2K, and COCO-Stuff and on SSIM results for ImageNet and DIV2K.
- Our methods: better on $L2_{ab}$ metric for DIV2K, Places365, and COCO-Stuff
- Semantic segmentation played an important role in enhancing the colorization results, and it helped our method improve the accuracy of the ab channels.

Qualitive Comparisions

❖ **SUCCESSFUL CASES**



Results were more vibrant and had more precise edges than the other methods. Moreover, the yellow color noise also was reduced in our ClasRegSoft versions comparison on RegSoft version.

Qualitive Comparisions

❖ **SOME FAIL CASES**

Ground-truth | **Input** | **Zhang** | **Iizuka** | **Larsson** | **RegSoft** | **ClasRegSoft** | **SegClasRegSoft**

My results met difficulties for colorization with incorrect colors, noise occurrences. These defects are similar to the results of Iizuka et al. and Larsson et al..

Project: VAE-Based Image Colorization

THANKS FOR LISTENING!

Waiting for question!